Taming Complexity in Search Matching: Two-sided Recommender Systems on Digital Platforms

Onkar Malgonde1, He Zhang2, Balaji Padmanabhan2, and Moez Limayem2

1Operations Management and Information Systems, College of Business, Northern Illinois University, 740 Garden Road, BH 328, DeKalb IL 60115

2Information Systems and Decision Sciences, Muma College of Business, University of South Florida, 4202 E. Fowler Avenue, CIS 1040, Tampa FL 33620

[omalgonde@niu.edu](mailto:omalgonde@niu.edu), [hezhang@usf.edu](mailto:hezhang@usf.edu), [bp@usf.edu](mailto:bp@usf.edu), [mlimayem@usf.edu](mailto:mlimayem@usf.edu)

**Abstract**

We study digital multi-sided platforms as complex adaptive business systems (CABS) where multiple sides have different and evolving objectives, preferences, and constraints. CABS are characterized by irreducible uncertainty which cannot be reduced by the traditional approaches of collecting and processing data. Irreducible uncertainty in the system gives rise to a complex search matching problem between agents and value enhancing transactions. This paper presents a recommender systems-based approach for taming the complexity by allowing agents to co-evolve and learn in the system. We propose a novel two-sided recommender system framework which considers emergence on both sides of the platform and adapts to the changing environment to influence agents. An agent-based simulation model is developed based on popular internet-based educational platforms to study this complex system and test our hypotheses. Our results show the value of a two-sided recommender system to tame complex search matching in platforms. We discuss implications for information systems and complexity science research.

**Keywords:** complex adaptive business systems, recommender systems, two-sided recommender system, digital platforms, complex search matching problem, agent-based simulation modeling

# INTRODUCTION

In this paper, we consider digital multi-sided platforms (hereafter, simply “platforms”) from the *market intermediary* stream (Thomas et al. 2014) as complex adaptive business systems (CABS). Specifically, we consider the *platform provider* model of organizing platforms where the platform mediates buyers’ and sellers’ interactions (Eisenmann 2008) and where the buyers, sellers, platform and platform owner(s) taken together form the CABS. In this model the platform seeks to attract new buyers and sellers and facilitate buyer-seller matches which lead to value enhancing transactions. Platforms also benefit by appropriating a fraction of the transaction value.

However, there may be both virtuous and vicious cycles here (Shapiro and Varian 1999). For instance, in ride-hailing (e.g., Lyft and Uber) platforms, during a virtuous cycle, drivers and ride seekers can find each other based on their search criteria, leading to an increase in drivers and ride seekers. Alternatively, in a vicious cycle, drivers and ride seekers cannot find each other due to lack of drivers and/or ride seekers. In general, better and more buyer-seller matches attract participants and create more transactions that enable virtuous cycles on the platform, but when the number of buyers and sellers increases, the shift could pose difficulties for participants in finding appropriate matches, possibly leading to abandonment or a reduction in activity on the platform. In this paper we view this search matching problem through the lens of complexity science which enables the design of a novel recommender system that helps tame the inherent complexity.

A complex problem “has many diverse parts that adapt and morph into new forms with every attempt to solve the problem. Finding an optimal solution to a complex problem is not feasible; the parts of the problem interact with each other in nonlinear ways, self-organize, and produce emergent macro-level behaviors that differ in scale and kind from the micro-level behaviors of the parts” (Tanriverdi et al. 2010, pp. 822-823). The general search matching problem on digital platforms has exactly these characteristics. Digital platforms have diverse participants, where diversity may manifest in the form of objectives, preferences, capabilities, and constraints. Diverse participants adapt in response to changes in their environment, which may include changes in product or service offerings, ratings, reviews, behavior rules, trends, competitors, and the platform’s governance mechanisms. Participants develop connections with others, in an often asymmetric manner, when one participant has a higher dependency on another (e.g., for information, product, services, or trends). Asymmetric relationships among participants lead to different outcomes. Such interactions are nonlinear, producing unpredictable outcomes for participants (Hardesty 2010). Such unpredictability gives rise to emergent behavior in response to the changing environment. Emergent behaviors further develop novel responses, leading to emergence[[1]](#footnote-2) on *both* sides of the platform.

Together, nonlinear interactions and emergence in the system lead to *irreducible uncertainty* for participants towards finding a match, resulting in a *complex* search matching problem. Associated with the future state of the system, irreducible uncertainty or fundamental uncertainty (Tanriverdi et al. 2010) is that which cannot be reduced by collecting and processing additional data, typically capturing what happened in the *past*. In the face of irreducible uncertainty, the focus shifts from reducing to *taming* the irreducible uncertainty. We define taming the irreducible uncertainty as the enablement of participants on the digital platform to co-evolve, learn, and improvise to improve agents’ performance in the system.

Existing solution approaches (filtering, tagging, one-sided recommender systems, among others) offer some help, but are quite limited, as we show later. Among these, recommender-systems-based approaches which nudge or guide micro-decisions, are natural mechanisms to consider in platforms. However, traditional recommender systems are “one-sided” algorithms which limit their ability to serve multiple agents in a platform. This paper proposes a two-sided recommender system framework which tames the irreducible uncertainty in the complex search matching problem by considering emergence on both sides of the platform and enables participants to co-evolve and learn to improve their performance.

We study this complex search matching problem in an applied context of newer internet-based educational platforms which offer massive open online courses (MOOCs). These educational platforms attract millions of students and hundreds of universities, all modeled as agents in this platform. Objectives, preferences, and constraints of students and universities differ and evolve with their experiences on the platform. Students and universities are connected to other participants and develop asymmetric dependencies which lead to nonlinear connections and introduce irreducible uncertainty in the educational platform with respect to their objectives, preferences, capabilities, constraints, and performance outcomes. We employ agent-based simulations to model this complex system and show how a two-sided recommender system can help tame the complex search matching problem.

Tanriverdi et al. (2010) proposed a renewed quest for IS research in CABS where IS applications should enable agents to co-evolve and learn in the system rather than collect and process data to maintain relevance when confronted with irreducible uncertainty. This paper (a) builds on this proposition by providing empirical evidence to show the limitations of existing IS solutions (one-sided recommendations in the case of platforms) and (b) proposes a new (two-sided) recommender framework which considers emergence in the system to tame irreducible uncertainty and improve agents’ performance in spite of the increasing uncertainty in the system. This research contributes to complexity science by introducing an adaptive IT-based mechanism which can tame the complex search matching problem. While multi-sided recommendations work for platforms such as the one we consider, it is likely that different types of complex systems need different methods for taming problems. There is little in the literature on how to tame complex systems in the face of irreducible uncertainty and this paper could lead to a renewed search for interesting solutions to tame various complex systems.

The paper is organized as follows. We first discuss the complexity concepts in digital platforms and the complexity of the search matching problem and review existing solution approaches. Identifying recommender systems as a potential solution approach, we discuss the existing one-sided recommender framework and its limitations. We then propose a novel two-sided recommender system and develop hypotheses related to how it tames the complex search matching problem. Moving then to the context where we are applying it here, we discuss the conceptual model of CABS and its instantiation in the internet-based educational platform and present the design of an agent-based simulation model. We then discuss the integration of a two-sided recommendation algorithm with the agent-based simulation design. We next present results and conclude with a discussion on limitations and implications for information systems, complexity science, and recommender systems research.

# THEORETICAL FOUNDATIONS

## Digital Multi-Sided Platforms as Complex Adaptive Business Systems

Building on Page (2009), Tanriverdi et al. (2010) note that in a CABS, elements show moderate degrees of (a) diversity (agents are neither too similar nor too dissimilar), (b) adaptation (agents learn to adapt to changing environments), (c) connectedness (agents are connected to each other), and (d) interdependence (agents develop dependency relationships with each other). Together, the digital platform and its owner(s), buyers, and sellers form the CABS (Figure 1). Appendix A maps the CABS concepts (Nan 2011) and four properties (Tanriverdi et al. 2010) required in a complex system for the digital multi-sided platform considered in this research.

|  |
| --- |
| Figure 1. Digital Multi-Sided Platform as a Complex Adaptive Business System |

Digital multi-sided platforms in the market intermediary stream facilitate transactions for diverse and geographically distributed participants on the platform (Hagiu 2006; Thomas et al. 2014), but they do not take ownership of the products and services transacted on the platform (Hagiu and Yoffie 2009). Examples of such platforms include Uber, Coursera, eBay, and Amazon. Owner(s) of the platform determine the architectural and governance mechanisms (Tiwana 2013) and participants choose to enter into the type and frequency of transactions.

## The Search Matching Problem and its Complexity

The platform has two primary motivations to facilitate value enhancing transactions. A high number of transactions induces positive network effects and feedback loops to attract new participants. Second, the platform derives value as a fraction of the overall transaction value. As the platform seeks to attract participants and facilitate transactions, it is harder for participants to find matches for completing value enhancing transactions (Parker et al. 2016). For example, on Amazon, a new parent may appreciate 5-10 sellers selling the same or different diaper brand. However, the same new parent will have to rely on other cues to decide if hundreds of sellers are selling the same or different diaper brand. Similar issues are reported in other digital platforms such as Airbnb (Solon 2018) where diverse offerings and buyer-types make the process of finding matches challenging.

Digital platforms are characterized by high complexity dynamics. Participants differ in their objectives, behaviors, resources, information and connections, and they have different adaptive mechanisms. Participants also develop and revise connections with other buyers and sellers as they enter into transactions and share information directly (messaging systems) or indirectly (reviews or ratings), creating a set of mutually dependent agents. In this process they also develop asymmetric dependencies. For example, a focal buyer develops greater dependency on sellers who offer exclusive products which interest the focal buyer. Similarly, sellers have greater dependency on specific segments of buyers whose interest aligns with the products and services offered by the sellers. Buyers have a greater dependency on other buyers who post reviews and rate items which they are interested in buying. Agents therefore develop multiple asymmetric dependencies on the platform, leading to nonlinear cause-and-effect interactions (Tanriverdi et al. 2010). Nonlinear cause-and-effect interactions produce different outcomes for similar actions performed by agents. For example, a focal buyer is negatively affected by a seller’s incompetence when the buyer has greater dependency on the seller. Another buyer with lower dependency on this seller may not be inconvenienced at all.

Each agent therefore senses, adapts, and develops novel responses to changes in the CABS. As agents differ in their capabilities, information, behaviors, and objectives, their responses in the system differ, but interact with responses of other agents, leading to unexpected outcomes which invoke new responses from the agents. Such emergence affects all agents in the system and invokes adaptive moves from them. Thus, nonlinear interactions lead to unexpected outcomes for agents’ actions, and any predictions are limited to the current state of the system and are less relevant in the next state. With all of these dynamics, it is impossible to exactly predict the future state given the current and previous states of the platform, or to determine optimal moves for each agent (where such predictions are necessary). Such unpredictability leads to irreducible uncertainty for each agent in the CABS.

In the face of irreducible uncertainty in the system, the goal should shift from reducing the uncertainty to *taming* the uncertainty inherent in the system. This requires participants to co-evolve with the changing dynamic landscape and continuously learn about the state of the system to perform actions which position them for relevance (Tanriverdi et al. 2010). However, agents cannot guarantee a favorable outcome, as nonlinear interactions between agents produce different outcomes and diverse agents possess different capabilities to co-evolve and learn.

Some research has highlighted related problems in different contexts. Asvanund et al. (2004) study peer-to-peer networks and find that with increasing network size, users are less likely to contribute to the network and impose additional costs on the network. Butler (2001), in the context of e-mail-based social structures, notes that large social structures are likely to free-riding and loafing, leading to a drop in member attraction and retention. Jones et al. (2004) find that in shared online spaces, where there is an information overload from increased interactions, individual users are likely to end active participation. Parker et al. (2016) discuss similar examples of search matching complexities in ride-hailing platforms (e.g., Uber) and information technology platforms (e.g., Covisint). The scale of search matching has also been shown to increase exponentially with every new participant (Tiwana 2013).

## Existing Approaches to the Search Matching Problem

Current approaches[[2]](#footnote-3) employed by the digital platform employ an information gathering and processing approach and can be broadly classified into two areas—human agent-driven and platform-driven. Human agent-driven solution approaches rely on a human-agent’s ability to process information and make decisions to achieve better outcomes. Participants categorize themselves based on some criteria (e.g., price, product characteristics, geographical location) which narrows the potential pool of matches. For example, Uber drivers list themselves in higher price categories that are requested by a segment of ride-hailers. On Amazon, buyers filter potential sellers based on specific price and/or ratings segments. The onus is on the human agent (buyer and seller) to use different criteria to seek matches, and the criteria often evolve with time and type of transaction. In platform-driven approaches, the digital platform provides tools and services to human agents to make decisions and achieve better outcomes. On today’s platforms, such tools and services include search engines, curation mechanisms, automated systems, and recommender systems. Buyers can seek value enhancing matches via search engines built into the platform by configuring search parameters to alter the match criteria (e.g., ratings, reviews, and popularity). Platforms use curation mechanisms and keywords to categorize products and services. For example, Apple classifies different applications under categories such as Games, Productivity, Health, and Finance, among others, to facilitate matching with potential buyers. The core strategy employed by these approaches is to limit the type of transactions buyers and sellers can participate by creating abstract groups (agents are forced to certain groups or agents self-elect to a group based on their participation objectives and preferences) within the different sides of the platform. Such static strategies, which are updated periodically or by the participant, have limited applicability to tame the irreducible uncertainty inherent in the CABS because (a) assignment and/or election to an abstract group locks the focal participant and subsequently affects search matches, (b) any change to the focal participant’s preferences and objectives requires an explicit action of reassigning the participant to an appropriate abstract group, (c) participants may be assigned/elected to abstract groups which may not fulfill the focal participant’s search matches, and (d) subtle changes in a participant’s preferences and objectives may not be captured.

In economics, Roth (1982) uses game-theoretic aspects to find that no procedure exists which will always provide a stable match (at which point agents no longer seek matches other than their existing match). Romanyuk (2016) and Kanoria and Saban (2017) find that full disclosure of information regarding agents’ preferences leads to rejections, thereby reducing matches. They identify partial disclosure of information as a surplus maximizing policy for platforms. Allon et al. (2012) show that operational efficiency may be detrimental to the platform’s efficiency unless complemented by communication between agents. Ashlagi et al. (2018) find that recommending matches and encouraging informative signals from agents result in stable matching. At a macro level, the search and matching theory provides a mathematical framework to identify optimal matches between different agents (Petrongolo and Pissarides 2001).

In this paper we present a novel data-driven two-sided recommender system framework to *tame* the complex search matching problem. When the problem is tamed, the agents’ performance fluctuations in the system are reduced, compared to when the agents operate in an un-tamed complex system environment. The two-sided recommender system design emphasizes learning emergence on different sides of the platform. Unlike the previously discussed approaches, a two-sided recommender system can support multiple sides of the platform and model highly granular platform dynamics within CABS by using an agent-based simulation model and data-driven solution mechanisms.

# RECOMMENDER SYSTEMS IN DIGITAL PLATFORMS

## Two-sided Recommender Systems

Recommender systems have primarily been used in platform ecosystems to serve one side (or a single agent type) and improve metrics which align with the objectives of the side being served. Recommender systems embody the canonical tasks identified for a recommender system, such as *find good items or predict an item’s relevance to a user* (Adomavicius and Tuzhilin 2005; Jannach and Adomavicius 2016). The literature on recommender systems has developed metrics such as precision, recall, coverage, and diversity to evaluate a recommender system’s efficacy (Shani and Gunawardana 2011). However, these metrics are limited to a user or set of users and do not consider the diversity of agents on platforms. Figure 2 illustrates the current conceptualization of recommender systems on platforms.

|  |
| --- |
| Figure 2. Current Conceptualization of Recommender Systems in Platforms |

Recent work in recommender systems on digital platforms has focused on improving the diversity of recommendations (Adomavicius and Kwon 2012), considering constraints before recommendations are served (Parameswaran et al. 2011), and considering reciprocal recommenders (people are subjects and objects of recommendations, such as job recommendations and online dating) (Pizzato et al. 2010; Xia et al. 2015). Reciprocal recommenders, in theory, address two sides of a platform. For example, they recommend dating partners (person X to person Y) or potential employees (recommend person A to recruiter B). However, these are currently quite limited in that (a) the recommendations are often based on preferences set forth by the user (person X should not be recommended to person Y if X does not satisfy Y’s preferences), and (b) the system considers symmetric individual goals across the users, i.e., find a suitable dating partner.

Godoy-Lorite et al. (2016), taking a complexity view, highlight the simplistic nature of existing recommendation approaches which do not consider complexity in the system. Current studies in recommender systems evaluate algorithm performance using a static strategy—data is split into training and testing datasets. Using a holdout sample from an earlier period to evaluate algorithm performance is limited since the strategy does not consider the speed with which complexity evolves on platforms. In this paper, we use an agent-based simulation model that can model the dynamics of digital platforms.

Importantly, traditional recommender systems are not designed to handle complex systems since they serve just one side and model a world in which the future is (more or less) like the past. However, in complex systems, participants are mutually dependent in seeking value enhancing transactions, and emergence on one side of the platform triggers self-organization and emergence on other side. For example, drivers on Uber (a ride-hailing platform) define their preferences (e.g., pooled or non-pooled rides) on the type of rides that they intend to honor. Such preferences trigger intelligent responses from buyers on the type of rides they seek, such as requesting non-pooled rides over pooled rides when in a hurry (Bindley 2018). When a recommender system focuses on one side of the platform (though utilizing platform-level data), the recommender system is beneficial to the side being served. However, a one-sided recommender system does not consider the effect of the other sides’ emergence and adaptation to the emergence of the side it serves. Thus, the recommendations served by a one-sided recommender system are limited to local patterns and preferences. For example, buyer-side recommender systems mine transaction data to determine buying patterns and serve contextual recommendations. However, they do not consider (a) emerging preferences on the seller’s side (e.g., inventory, competition) in real time, and (b) the effects of emergence in buyers on sellers (e.g., trends, profitability). These issues have been highlighted in recent discourse in recommender systems (Jannach and Adomavicius 2016) and complexity literature (Godoy-Lorite et al. 2016). Complexity researchers have also highlighted the narrow focus considered in recommender systems. Godoy-Lorite et al. (2016) present a collaborative filtering model that, unlike prior work, does not assume that a focal user belongs to a single group of users. Instead, Godoy-Lorite et al. (2016) allow a focal user to belong to multiple groups, and thus their algorithm achieves better accuracy than other algorithms.

|  |
| --- |
| Figure 3. Proposed Conceptualization of Recommender Systems in Platforms |

Many of these limitations in the current recommendation system framework are due to a lack of recognition that platforms are complex sociotechnical systems where heterogeneous agents are interconnected, differ in their objectives, motivations, and requirements, and continuously co-evolve. Figure 3 illustrates the design of the proposed two-sided recommender system framework which learns the emergence on multiple sides of the platform and adapts to changes on the platform; such an approach moves away from merely collecting and processing data to predicting the future state of the system and taking a more holistic view. Individual sellers and consumers can access a subset of platform data (shown for each seller in Figure 3) which consists of transactions in which they participated. In comparison, recommendation frameworks draw upon platform-level data which tracks all transactions (Figure 2 and Figure 3). However, as our experiments using agent-based simulations show, accessing and analyzing more data does not result in better decisions in CABS because intelligent agents develop new behaviors that may render current solutions irrelevant and forecasting infeasible.

In our approach, the two-sided recommender system is itself an agent in the CABS whose objectives, behavior, and rules are defined by the platform owner(s). Although the two-sided recommender system is connected to participants on the platform, it (like other agents) does not control participants. Instead, the recommender *influences* participants who may still choose to disregard the recommendations. While traditional recommender systems also take this perspective, embedding the two-sided recommender itself as an agent in a complex system is an important aspect of understanding the influence such a system has on emergent behavior.

Considering emergence on different sides of the platform allows the two-sided recommender system to enable other agents to learn and co-evolve in the CABS to *improve* their performance outcomes. Contrast this to the data processing and optimizing view of existing solution approaches which aims to reduce the irreducible uncertainty and predict the future state of the system to position agents for relevance. Despite considering emergence on the platform, the two-sided recommender system cannot *guarantee* superior performance outcomes for agents in a complex adaptive business system.

## Hypotheses

Digital multi-sided platforms are complex adaptive business systems where agents face a performance landscape which is characterized by local and global optima which change over time (Lee et al. 2010; Tanriverdi et al. 2010). In such an evolving landscape, agents can achieve better outcomes by repositioning themselves by co-evolving with the landscape (Wiggins and Ruefli 2002, 2005) in comparison to the strategy of accessing, analyzing, and predicting the system’s state (Tanriverdi and Lim 2017). IT-based solution mechanisms, specifically recommender systems, can augment agents’ ability to co-evolve and learn in the complex adaptive business system (Merali et al. 2012).

To evaluate the benefit of recommender systems in taming the complex search problem, we consider three variants: no recommender system, a one-sided recommender system, and a two-sided recommender system. The no recommender system scenario is considered a base case where agents interact in the CABS without any recommendations. A one-sided recommender system relies on platform-level data to learn macro-level trends and patterns, and thus emergence, specific to the side it serves; i.e., it augments learning and co-evolution of agents on that side. However, the one-sided recommender system is incapable of fully learning the emergence on the side it serves because there is emergence on other side of the platform as well. Such an approach to collecting, processing, and learning specific aspects of the platform limits the one-sided recommender system’s ability to enable participants to co-evolve with the complexity in the environment. By modeling emergence on both sides of the platform, the two-sided recommender system augments agents to co-evolve better, thereby taming the complex search matching problem. This results in reduced fluctuations in agent-performance compared to an untamed complex system environment.

***Hypothesis 1a:*** *A one-sided recommender system will tame the complex search matching problem better than using no recommender system.*

***Hypothesis 1b:*** *A two-sided recommender system will tame the complex search matching problem better than using either no recommender or a one-sided recommender system.*

Taming the irreducible uncertainty will allow participants to co-evolve and learn from their environments. Co-evolving participants are better able to position themselves for relevance in complex systems (Merali et al. 2012; Tanriverdi et al. 2010; Tanriverdi and Lim 2017). As the two-sided recommender system serves recommendations, participants can exploit the temporary advantage before co-evolutionary moves by other participants. The two-sided recommender will update its recommendations in subsequent periods to reflect changes in the environment. However, superior performance cannot be guaranteed in CABS because participants develop novel responses which may not be immediately learned by the recommender system. Improved performance may relate to better matches, completed objectives, or higher profits for participants. Together, we introduce *fitness* of a participant to encompass the current state of the agent which represents the overall ability of the agent to continue participation in value enhancing transactions on the platform. Low fitness leads to an exit from the platform whereas high fitness leads to continued activity on the platform.

***Hypothesis 2a:*** *With increasing uncertainty on the digital platform, a one-sided recommender system will improve the average fitness of buyers and sellers in comparison to no recommender system.*

***Hypothesis 2b:*** *With increasing uncertainty on the digital platform, a two-sided recommender system will improve the average fitness of buyers and sellers in comparison to either no recommender system or a one-sided recommender system.*

In summary, the CABS of a digital multi-sided platform gives rise to a complex search matching problem which cannot be addressed by the traditional approach of collecting and processing data. In light of this research gap, the design of an IT-based solution approach requires consideration of the causes of complexity in the CABS (McKelvey et al. 2015) and adjusts the assumptions and logic of IT-based solution approaches (Tanriverdi et al. 2010). Existing one-sided recommendation frameworks fail to consider the complexity in CABS. This paper addresses this gap by incorporating the causes of complexity in the design of the two-sided recommendation framework artifact to tame the complex search matching problem in digital multi-sided platforms and illustrate the role of IT-based artifacts in complex systems.

# CONCEPTUAL MODEL OF AN INTERNET-BASED EDUCATIONAL PLATFORM AS A COMPLEX ADAPTIVE BUSINESS SYSTEM

As a case study to illustrate the efficacy of a two-sided recommender system to tame the complex search matching problem and improve agent fitness on the platform, we consider newer Internet-based, multi-sided educational platforms such as Coursera and Udacity, which provide massive open online courses (MOOCs). Universities offer courses in these platforms that a student can audit or take to complete a certification or specialization. Geographically distributed students with different backgrounds and interests can access these courses in different formats.

Over the last decade, educational platforms have been lauded for making world-class education available to anyone with an internet connection and are key to renewed data-driven efforts to study learning in humans.[[3]](#footnote-4) Despite these benefits, recent criticism of MOOC platforms includes low completion rates, weak discussion board participation, lack of detailed feedback, and excess effort requirement, among others (Adams 2013; Ubell 2017). Recent work in recommender systems on such educational platforms has focused on developing systems that can recommend discussion boards (Yang et al. 2014), similar participants (Prabhakar et al. 2017), videos addressing conflicts in discussions (Agrawal et al. 2015), and relevant courses (Li and Li 2017). These studies follow the one-sided recommendation framework; to our knowledge, there is no prior work which considers the irreducible uncertainty of search matching in such educational platforms. Consequently, these studies focus on the recommender system’s efficacy to address a specific problem for the side (often, the student-side) being served. Table 1 lists agents and their objectives in this educational platform. Figure 4 illustrates the conceptual model of the educational platform as a complex adaptive business system.

|  |  |
| --- | --- |
| **Agent** | **Objective** |
| Student | Complete representative courses/modules from her area(s) of interest, audit or complete certification or specialization, improve grades and receive good feedback, increase interactions |
| University | Offer courses in area(s) of interest, attract diverse students, maintain utilization threshold, improve ratings |
| Educational Platform | Facilitate course offerings, registration, grading, and interactions |
| Platform Owner(s) | Orchestrate platform to attract new students and universities, increase registrations and offerings |
| Recommender System | Increase student and university engagement (recommend activities on the platform), recommend courses to audit or complete certification or specialization, recommend course offerings to maintain utilization threshold |

Table 1. Agent Objectives in Educational Platform

|  |
| --- |
| Figure 4. Conceptual Model of Internet-based Educational Platform |

Universities’ motivations to offer courses have been credited to multiple goals including extending reach and access, building and maintaining the brand, reducing costs or increasing revenues, improving educational outcomes, and improving innovation and conducting research on teaching and learning (Hollands and Devayani 2014). Given the massive number of enrolled students in each class, courses use innovative ways to deliver content (videos and interactive quizzes) and grade student submissions (auto grading, peer grading, teaching assistants, or a combination of these), knitting together education, entertainment, and social networking (Pappano 2012). Courses are periodically offered throughout the year, and prerequisites are suggested but not enforced. A variety of heuristics such as faculty availability, demand, interactions with other universities, and recommendations from the platform (if any) are used to decide course offerings. Universities may develop asymmetric connections (based on reputation and experience) with other universities via interpersonal channels to share experiences and best practices. Universities also show adaptive behavior in response to changes in trends and other universities’ offerings. These asymmetric connections and adaptations introduce emergent system-level behavior on the university side which influences new and existing students and universities. Further, a focal university’s outcome is dependent on its actions and other agents’ actions on the platform because any adaptation (e.g., changes in offerings, course structures) by the focal university triggers adaptive responses from other universities and students (e.g., reviews, ratings, offerings).

Students’ motivations to participate in MOOCs have also been accredited to multiple goals, including to satisfy prerequisites or gain credits towards an academic program, refresh key concepts, prepare for standardized tests, and identify potential universities for future degree enrollment (Zhenghao et al. 2015). Students find multiple universities offering competing courses in an area of interest. Course reviews and ratings from other students provide vital cues for the focal student to decide for which courses she would register. Also, students can try different courses, continue with ones that they find valuable, and skip to relevant content in a course or complete the course at their pace. In addition to these platform offerings, students develop asymmetric dependencies with other students (seeking information, grades, or ratings) and universities (courses, certifications, specializations, faculty, or brand). Students also adapt to their changing environment (grades, course offerings, ratings, reviews, job market, trends) by adjusting their resources (time spent on a course), taking new courses, and dropping existing courses. As students develop asymmetric dependencies and adapt to the changing environment, they introduce emergent system-level behavior on the student side which influences new and existing students and universities. Finally, a focal student’s outcome is dependent on her actions and other agents’ actions on the platform because any adaptation (e.g., registration, assignment, peer grading) by the focal student triggers an adaptive response from other agents (e.g., grades, offerings, discussion board comments).

The educational platform allows universities and students to participate in transactions (offer and register for courses to satisfy their participation objectives). As is the case in most platforms, the educational platform derives value by extracting a fraction of the overall transaction value (students pay for certifications and specializations). While universities utilize their transaction data, platforms (and any associated recommender system) can draw upon platform-level data. The educational platform provides standardized interfaces and procedures for universities to offer their course content and manage student interactions. Similarly, the platform provides search capability and recommendations to students about potential courses and actions based on their prior transactions or activities from similar students. The platform owner(s) determine the architecture and governance mechanism of the educational platform (Tiwana 2013). These mechanisms aim to attract a diverse set of students and universities to the platform and increase student and university participation across different activities. The challenge for the platform is to maintain and grow a critical mass of universities offering courses and students paying, registering, and completing courses—a virtuous cycle. If universities and students have negative experiences (e.g., low registration, high cost, lack of quality courses, lack of feedback) on the platform, they will either abandon the platform or reduce activity on the platform—a vicious cycle (Shapiro and Varian 1999).

Adopting the perspective of the platform owner(s), we consider the recommender system’s objective as maximizing student and university matches while considering their constraints, objectives, and preferences. The proposed two-sided recommender system draws on platform-level data and collectively considers the diverse agents’ objectives to serve recommendations. Recommendations to universities may include course offerings, areas of interest, and frequency of course offerings. For students, recommendations may include courses, areas of interest, specializations, and peers for grading. It is important to note that the two-sided recommender system should also consider, in real-time, the impact of recommendation acceptance and rejection on the platform. Appendix B provides a concise presentation of the conceptual model of this internet-based education platform as a complex adaptive business system. To identify common themes and practices, we studied leading educational platforms and identified components that are most relevant to our study and adopted a parsimonious approach to replicate a multi-sided educational platform. In what follows, we will operationalize the conceptual CABS model discussed in this section by presenting an agent-based model which incorporates the two-sided recommender system.

# TWO-SIDED RECOMMENDER SYSTEMS IN EDUCATIONAL PLATFORMS

## Agent-Based Model Design

Agent-based simulations have been widely adopted in CABS (Epstein and Axtell 1996) and IS research (Nan 2011; Nan and Tanriverdi 2017) as they enable researchers to parsimoniously model complex phenomena and study macroscopic emergent behavior (Amaral and Uzzi 2007). Agent-based simulation modeling in CABS has used a string of symbols (e.g., 10110001) to represent an agent and its attributes. Each string element is randomly assigned a value from a predetermined range, such as (0, 1] or from a set of possible values such as {0, 1, -1}. An arbitrary number of symbols (string elements) is chosen to represent each agent. In the model validation phase, the number of symbols in a string is varied to test the effect on results. String-based representation schemes also provide efficient interaction capabilities to the researchers in the simulation model while abstracting real-world details. A popular choice among researchers to implement such agent-based models is object-oriented programming where an object represents an agent and the object’s variables represent an agent’s attributes.

In our agent-based simulation model, we use object-oriented programming. However, we define variables and data structures to identify an agent’s attributes instead of the string-based representation technique because our experiments require data structures to archive past agent states—student activities like registrations, searches, interactions, or social networking; university activities like offerings, prerequisite structure, social networking; and recommender system learning trends, filtering future activities from agent activities. Although this implementation differs from traditional string-based simulation modeling, it preserves the logical underpinnings by using a random assignment of initial values, a clear definition of interaction logic, an internal clock to represent time, and simple behavioral rules. To identify agent characteristics, objectives, and behavior rules, we studied leading internet-based educational platforms such as Coursera and Udacity. We also surveyed literature discussing students’ and universities’ motivations, objectives, and behaviors, on internet-based educational platforms (Adams 2013; Agrawal et al. 2015; Konnikova 2014; Zhenghao et al. 2015). While this approach does not capture all the complexities of reality, it represents a parsimonious approach to capturing a simplified picture of reality (Nan 2011; Nan and Tanriverdi 2017) and allows us to study the effect of different recommender system frameworks on taming the complex search matching problem.

Figure 5 illustrates the abstract agent-based model design. Each student, university, and course is represented by variables and data structures where values may be integers (e.g., number of registered courses), unit intervals (e.g., fitness, learning rate), or data structures such as key-value pairs (e.g., time of course completion, areas of interest, current courses, grades, course ratings). Appendix C provides definitions, operationalization, and possible values for attributes of students, universities, and courses.

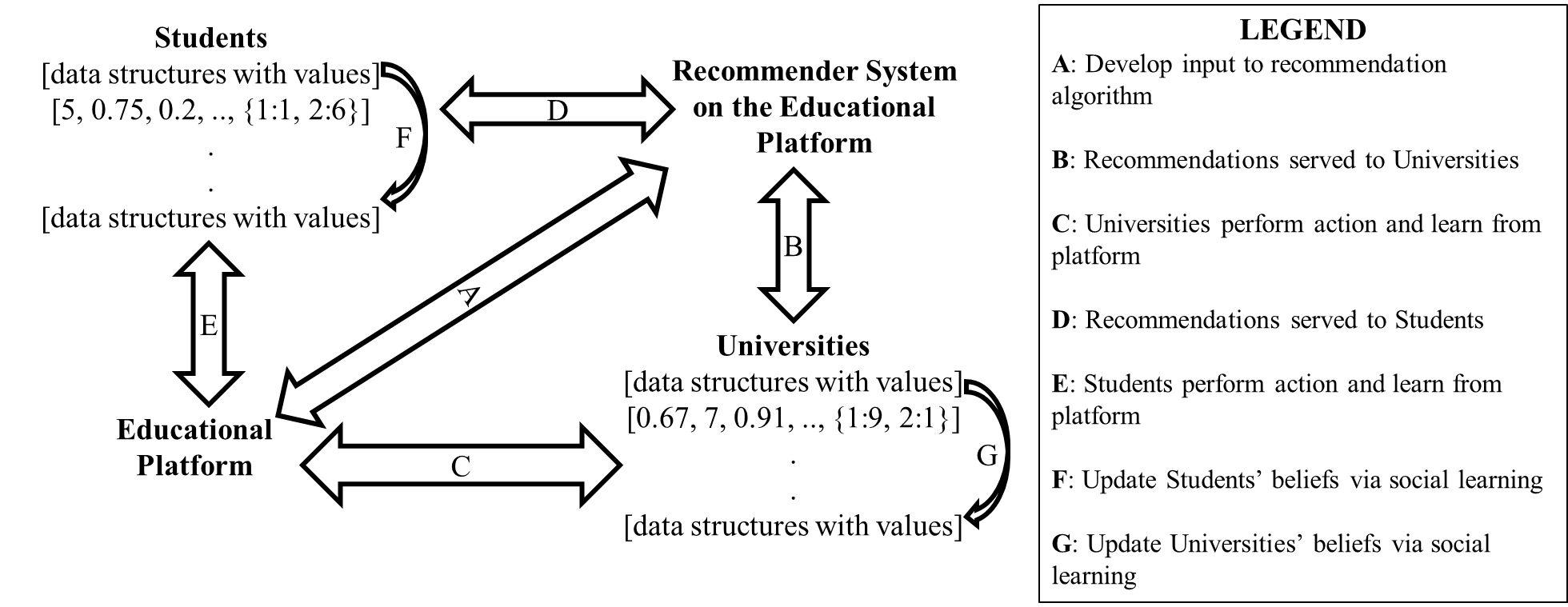


Figure 5. Design of the Agent-Based Model

The recommender system will utilize platform-level data to develop input for its behavioral rules (step A) and serve recommendations to active universities based on their internal state (step B). If a university accepts the recommendations, these are included in the input set of the focal university’s behavioral rules and the focal university executes its behavioral rules to perform an action (offer courses, participate in discussion board, grade) on the platform (step C) and interact with other universities in their network (step G). The recommender system influences students’ input sets for behavioral rules by serving recommendations (step D). As students execute their behavioral rules, they perform actions (registration, submission, peer grading, participating in discussion boards) on the platform (step E) and interact with other students in their social network (step F). For tractability we assume that the architectural and governance mechanisms set by the platform owner(s) are stable. The recommender system incorporates the objectives of the platform owner(s) and interacts with other agents on the platform. Also, platform owner(s) can adjust recommender systems’ parameters to embody any change in their objectives. Appendix D summarizes the agent-based model design.

## Agent Objectives and Fitness

|  |  |
| --- | --- |
| **Agent** | **Objective** |
| Student (Type 1) | Audit representative courses from area(s) of interest; improve grades |
| Student (Type 2) | Earn certification from representative courses from area(s) of interest; improve grades |
| Student (Type 3) | Specialize in area(s) of interest; improve grades |
| University (Type 1) | Offer audit courses in area(s) of interest; improve utilization rate; improve course ratings |
| University (Type 2) | Offer certification courses in area(s) of interest; improve utilization rate; improve course ratings |
| University (Type 3) | Offer specialization courses in area(s) of interest; improve utilization rate; improve course ratings |
| Recommender system | Enhance search matches between students’ and universities’ courses such that students’ and universities’ objectives and preferences are met; attract new students and universities; improve students’ and universities’ experience and interactions |
| Platform | Facilitate interactions within students and universities and between students and universities |

Table 2. Agent Objectives in the Agent-based Simulation Model

Table 2 identifies agents and their objectives in the agent-based simulation model. Each agent (student, university, platform, recommender system) is operating to maximize its objective. For a focal student, the objective is to complete courses from her area(s) of interest and achieve good grades on submissions. However, students differ in their objectives because of their types, area(s) of interests, and grades. For universities the objective is to offer courses with high ratings and maintain utilization rates to offset costs. However, universities differ in their objectives because of their types, ratings provided by students, and utilization rates due to differing popularity of their courses. For the recommender system, the objective is to enhance search matches for students and courses offered by universities such that students’ preferences are met and universities’ utilization criteria are maintained. In considering students’ preferences, the recommender system also considers learned patterns which will position a focal student to achieve better grades (suggested or learned prerequisites). Similarly, the recommender system also considers existing offerings (for utilization rates) and action possibilities (improve ratings) when making recommendations to the universities. The recommender system also incorporates the platform owner(s)’ objectives of attracting new students and universities and improving students’ and universities’ experiences. Finally, the objective of the platform is to offer an environment that facilitates student and university matches.

As agents participate on the platform to complete their participation objectives, we introduce *fitness* as the measure of an agent’s ability to continue participating on the platform and achieve those objectives. Initially, each student and university is assigned[[4]](#footnote-5) a uniformly random fitness value from the unit interval, and at each period the fitness is updated (increase or decrease by a random value) based on the experiences of the focal agent (student or university). We choose to assign an initial fitness value because of the contextual factors in educational platforms. Specifically, newer internet-based educational platforms are characterized by volatile incoming and attrition rates. Students may leave the platform any time—after browsing offered courses, completing a course module, partially completing a course, completing some certification courses or specialization from an area of interest; universities may leave the platform due to constraints from administration, faculty, or staff. Therefore, a student or university with an initial fitness of 0.2 has a greater probability of exiting the platform before entering the system. As the agent pursues its objective, the initial value of 0.2 is incremented or decremented based on the agent’s experiences, thereby increasing or decreasing the probability of an exit from the platform. The *fitness* of an agent (student or university) is an intrinsic emergent property which cannot be directly accessed or manipulated by other agents (students, universities, platform, or recommender system).

|  |
| --- |
| **Change in student’s fitness 🡪 *f* (signal from social network, grades, objective’s progress)**   * If average weighted *fitness* of *student(i)*’s social network is greater than random number   + *fitness(student(i), t)* = *fitness(student(i), t-1) + δ* where *δ = N(fitness(student(i), t-1), 3)* * Else   + *fitness(student(i), t)* = *fitness(student(i), t-1) - δ* where *δ = N(fitness(student(i), t-1), 3)* * If *student(i)’s* grade exceeds expectation   + *fitness(student(i), t)* = *fitness(student(i), t-1) + δ* where *δ = N(fitness(student(i), t-1), 3)* * Else   + *fitness(student(i), t)* = *fitness(student(i), t-1) - δ* where *δ = N(fitness(student(i), t-1), 3)* * If *student(i)*’s objectives are being met   + *fitness(student(i), t)* = *fitness(student(i), t-1) + δ* where *δ = N(fitness(student(i), t-1), 3)* * Else   + *fitness(student(i), t)* = *fitness(student(i), t-1) - δ* where *δ = N(fitness(student(i), t-1), 3)* |

Table 3. Pseudocode for Change in Students’ Fitness

For a student, a change in her fitness (Table 3) is dependent on signals from the student’s social network, grades, and objective’s progress. A focal student’s fitness is incremented if she receives positive signals from other connected students. Each student is connected (nonuniform weights) to a maximum of ten other active students. We increment the focal student’s fitness if the average of the weighted fitness (connection weight\*connected student’s fitness) is greater than or equal to a randomly generated unit interval; if not, we decrement the focal student’s fitness. Different weights represent the asymmetric dependencies for the focal student and the sharing of intrinsic fitness represents the flow of information about the connected student’s experiences on the platform. Second, each registered student receives grades for submissions. If the grades exceed the randomly generated expectation, the focal student’s fitness is incremented. Finally, the focal student’s fitness is incremented if the student’s registration objectives (complete representative course, certification, specialization) are met. In cases where these conditions are not met, we decrement the focal student’s fitness for each condition.

|  |
| --- |
| **Change in university’s fitness 🡪 *f* (signal from its social network, utilization goal, course ratings)**   * If average weighted *fitness* of *university(i)*’s social network is greater than random number   + *fitness(university(i), t)* = *fitness(university(i), t-1) + δ* where *δ = N(fitness(university(i), t-1), 3)* * Else   + *fitness(university(i), t)* = *fitness(university (i), t-1) - δ* where *δ = N(fitness(university(i), t-1), 3)* * If *university(i)*’s utilization meets or exceeds its utilization goal   + *fitness(university(i), t)* = *fitness(university(i), t-1) + δ* where *δ = N(fitness(university(i), t-1), 3)* * Else   + *fitness(university(i), t)* = *fitness(university(i), t-1) - δ* where *δ = N(fitness(university(i), t-1), 3)* * If *university(i)*’s average course ratings are greater than random rating   + *fitness(university(i), t)* = *fitness(university(i), t-1) + δ* where *δ = N(fitness(university(i), t-1), 3)* * Else   + *fitness(university(i), t)* = *fitness(university(i), t-1) - δ* where *δ = N(fitness(university(i), t-1), 3)* |

Table 4. Pseudocode for Change in Universities’ Fitness

For universities, a change in fitness (Table 4) is dependent on signals from its social network, utilization rates, and course ratings. For a focal university, fitness is incremented if it receives positive signals from other connected universities. Also, the fitness is incremented if its utilization goals across all offered courses are met. For each completed course, we compute the utilization rate by associating different weights to the type of student: a weight of 0.5 for Type 1 student, 0.75 for Type 2, and 1 for Type 3, to represent the revenue. The utilization goal [0, 1] is met if the utilization rate is greater than or equal to the utilization goal. Finally, the university’s fitness is incremented if its courses receive high ratings (average ratings exceed the uniformly random rating). For any conditions which are not met, we decrement the focal university’s fitness for each condition. It is important to note that we have not included financial metrics in the computation of agents’ fitness. We exclude financial metrics in our simulation model because of the still somewhat fluid and emerging nature of the business models in MOOCs, ranging from free, nominal-charge and full-blown tuition scenarios that are all playing out in the markets now. Incorporating this aspect will be an interesting direction for future work.

## Agent-Based Simulation Procedure

Figure 6 illustrates the simulation flowchart. We consider each period as one week where new students and universities are added to the pool of potential participants. Students and universities are activated after they receive positive feedback from their active peers on the platform. Upon activation, students and universities perform actions such as course registration and offering courses, assignment submission and grading, peer grading, and discussion board participation. Actions lead to outcomes and determine changes to individual fitness. We implement three simulation settings—no recommendation, separate one-sided recommender systems for students and universities, and a two-sided recommender system for students and universities. Appendix E provides the pseudocode and describes in detail the agent-based simulation.



Figure 6. Agent-based Model Simulation Flowchart

It is important to note that complexity emerges naturally[[5]](#footnote-6) in the agent-based simulation model as agents follow simple behavioral rules and interact with other agents. We identified broad types of students (interested in registering for audit, certification, and/or specialization) and universities (offering audits, certifications, and/or specializations) from the empirical context. While this allows us to model similarities in agents, variables characterizing internal states and behavioral rules employed by these agents are different. For example, the initial internal states of students (fitness, probabilities, constraints, social network, and learning rates) and universities (fitness, probabilities, constraints, and social network) are randomly determined. Further, agents develop different input sets to their behavioral rules as their individual coverage of the overall system is limited. Thus, broadly similar agents have differing outcomes (choices, grades, interactions) in the simulation. Together, the dynamics simulate a system that cannot be predicted and operates far from equilibrium.

## Two-Sided Recommender System

In this section, we discuss the recommendations served by the one-sided and two-sided recommender systems in our agent-based simulation model. Table 5 lists the types of recommendations each agent (student and university) is served in the simulation model. For the one-sided recommender system recommending to students, each active student receives recommendations on courses before registration, on peers before peer-grading assignments, and on discussion boards. For a focal student, the one-sided recommender system will recommend courses for registration for which the focal student is eligible and with a randomly increasing difficulty level. Such a recommendation allows students to complete certifications and/or specializations in their areas of interest. For peer-graded assignments and discussion board assignments, a one-sided recommender system will recommend peers from the focal student’s network (if they are registered for the same course) and students with similar areas of interest. We assume that students from the focal student’s network and area of interest will have a higher propensity to provide extensive feedback and improve the focal student’s experience. The one-sided recommender system will recommend to universities courses from an area of interest and with increasing difficulty, allowing the focal university to develop certification and/or specialization programs. Also, the one-sided recommender system recommends reoffering courses for universities with threshold utilization rates.

|  |  |
| --- | --- |
| **Agent** | **Two-sided Recommendations** |
| Student | Course registration, specialization, peers, discussion boards |
| University | Course offerings, specializations, offering times |

Table 5. Agent Recommendations

In the case of the two-sided recommender system, a focal student is given recommendations for eligible courses of increasing difficulty and courses from the area(s) of interest of the focal student. Such recommendations position the student to complete certifications and/or specializations and allow the recommender system to move demand to courses which have lower utilization. Similar to the process in the one-sided recommender system, the two-sided system recommends students from the focal student’s social network and area(s) of interest for peer-grading assignments and discussion boards. Such recommendations improve the focal student’s experience on the platform with better feedback from peers.

For universities, the two-sided recommender system will recommend offering courses from area of interest with increasing difficulty level and sufficient demand. Courses in an area of interest with increasing difficulty allow the focal university to offer certifications and/or specializations in the area of interest whereas sufficient demand may allow the focal university to achieve threshold utilization rates. Further, courses with threshold utilization rates are recommended to be reoffered by the two-sided recommender system. Also, recommendations are served to develop specializations based on the utilization rates of the focal university.

The one-sided recommender system follows the extant framework in recommending actions to students and universities. Historical data on the platform is analyzed to identify recommendations which may improve outcomes for focal students and universities. The recommender system’s suggestions to students does not interact with the recommender system for universities, nor does it consider emergence in the university side of the platform. In contrast, the two-sided recommender system utilizes platform-level data and learns emergence across the different sides of the platform to develop recommendations. Specifically, recommendations for the focal student are developed by considering the universities’ objectives, in addition to the objectives and preferences of the focal student, and vice versa. Recommendations served by considering the emergence in the platform enable students and universities to co-evolve in the system. Recommendations served by the two-sided recommender system consider the impact on the other side of the platform of recommendations served to the first side. This allows the recommender system to incorporate the consequences of agents’ actions in recommendations and subsequently position students and universities to achieve better performance outcomes.

In this research we focus on highlighting the complexity of the search matching problem and showing the value of a two-sided recommender framework. Hence we instantiate the two-sided framework with a simple heuristic-based algorithm. Adopting the design science paradigm (Hevner et al. 2004), the two-sided recommender system develops a technology-based solution to a relevant, timely, and complex problem of search matching on digital platforms. Table 6 maps the design science research guidelines to the work presented in this paper. To evaluate the proposed two-sided recommendation framework, we simulate an internet-based educational platform using agent-based modeling. The two-sided recommender framework is instantiated and compared with instantiation of extant frameworks—a one-sided recommender—and no recommender system. Our experiments (next section) show that the two-sided recommender system outperforms the one-sided recommender system in taming the complex search matching problem in CABS like internet-based educational platform and improves the overall fitness of agents in the system.

|  |  |
| --- | --- |
| **Guideline** | **Research** |
| Design as an Artifact | The artifact designed in this paper is the two-sided recommender framework. |
| Problem Relevance | This technology-based artifact is designed and instantiated to address a relevant, timely, and complex problem of search matching on digital platforms. The artifact is instantiated in an internet-based educational platform. |
| Design Evaluation | The efficacy of the design artifact is demonstrated by executing its instantiation and comparing it with existing approaches using an agent-based simulation model. |
| Research Contributions | For information systems and recommender systems literature, Figure 2 and Figure 3 provide a clear distinction between the current recommender systems’ conceptualization and the proposed conceptualization. For complexity science, the proposed artifact provides an IT-based tool to tame complex search matching problems in CABS. |
| Research Rigor | Building on the complex systems and recommender systems literature (together, the knowledge base), this work proposes a design artifact to tame the complex search matching problem. The design artifact is instantiated, and its efficacy is demonstrated using simulation experiments. |

Table 6. Adoption of Design Science Research Guidelines

## Complexity in the Agent-based Simulation Model

In this subsection we show the complexity in the agent-based simulation model which does not stabilize over time. A simulation run was performed for each recommendation system framework with baseline parameter values for 5000 periods. For each period, the average fitness of students and universities was recorded. We adopt the approach by Nan and Tanriverdi (2017) to show complexity in our agent-based simulation model. Nan and Tanriverdi (2017) fit linear and nonlinear models to the simulation data and compare Akaike Information Criterion (AIC) values to ascertain nonlinearity of the agent-based simulation model. Specifically, better fit of a nonlinear model shows a nonlinear effect of agents’ efforts on their performance outcomes. In addition, such an approach is appropriate in simulation studies where agents’ initial parameters are initialized from a distribution. For instance, each student’s fitness is assigned using a uniformly random distribution. Over time, we see that the average fitness of students is around 0.5. This is an obvious result based on the distribution of the function used to generate fitness values. If the agent-based simulation model did not exhibit hyper turbulence and achieved equilibrium, we would consistently see average fitness values around the 0.5 value. By fitting a linear and nonlinear model, we can ascertain whether there are changes in the average fitness values and subsequently conclude that complexity is present.

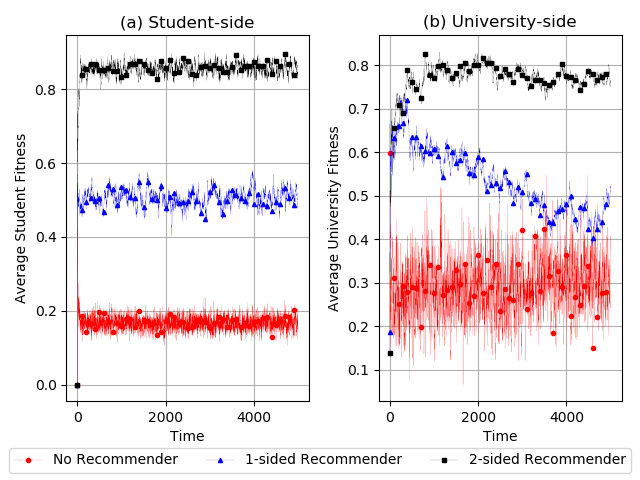


Figure 7. Simulation Run with 5000 periods

We use the data presented in Figure 7 (the figure shows a simulation run with 5000 periods; other analyses are conducted with simulation runs of 1000 periods) to fit a linear and nonlinear model. Specifically, each of the three models (linear, quadratic, and cubic) is fitted to each variant (no recommendation, one-sided, and two-sided) of an agent (student and university). For an agent’s variant, the best fitting model is selected based on AIC values. For example, a cubic model (AIC = -24554.72) is the best fit for the no-recommendation variant in comparison to a quadratic (AIC = -24526.1) or linear model (AIC = -24477.78). For each variant of agents, the cubic model was a better fit. We did not test additional nonlinear models as they may overfit the data. This analysis supports the nonlinear effect of agents’ actions on their performance outcomes over 5000 periods. To test the smaller periods, we divided the dataset into different bins of size 10, 25, 50, 75, 100, 150, and 200, and computed AIC values for each bin. For instance, 5000 data points from the no-recommendation system of student side were divided into 500 bins with 10 data points each. For each bin, we fit a linear and nonlinear model and compute AIC values. Using the Wilcoxon signed-rank test (assess if population means differ), a nonparametric test, we compare the mean of the AIC values across the 500 bins. For each window, the mean AIC values for the nonlinear model were statistically significant and lower in comparison to the linear model. This analysis was repeated with university-side data with similar results. Thus, we conclude that the complexity is present in the agent-based model for shorter and longer periods.

# RESULTS

## Taming Complex Search Matching in Digital Platforms

Taming the complex search matching problem enables agents to co-evolve in the CABS and improve their performance. When the search matching problem is tamed, the fluctuations associated with performance outcomes of agents are lower. To study the efficacy of the proposed recommender system framework, we extend the analyses from section 5.5. We consider the data from the simulation run with 5000 data points. The best fit model for each recommender system variant is identified (a cubic model was the best fit for all recommender system variants). Further, we compare the Akaike Information Criterion (AIC) values of best fit models across multiple data windows. Specifically, we divide the data (5000 data points for the average student and university fitness for each of the 3 recommender system variants) into 16 bins where each bin consists of 300 data points. For data in each bin, we fit the best model for the entire dataset and compute AIC values.

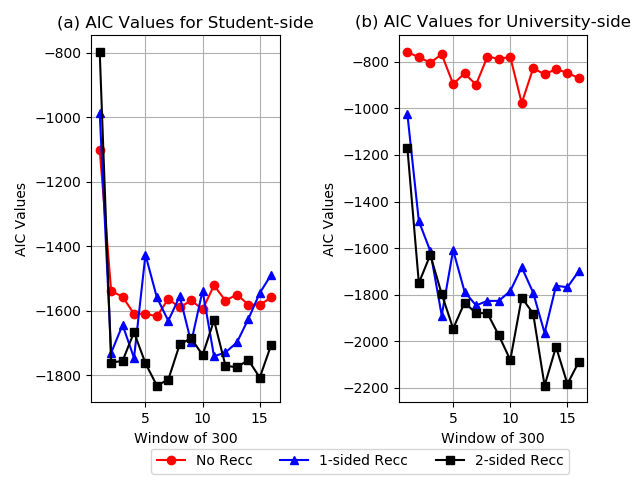


Figure 8. AIC Values for Window Size of 300

Figure 8(a) and (b) plot 16 AIC values for each data bin and the recommender system variant for the student and university sides, respectively. We see that the AIC values associated with the two-sided recommender system variant are often lower for the student and university sides. However, the AIC values for the one-sided recommender system are intermittently lower than no recommender system for the university side and consistently lower than no recommender system for the student side.

A lower AIC value for any of the 16 windows represents a better model fit for the variant and lower error with respect to the best global model. As we are fitting the best global model (cubic) to different windows of the data, we quantify the taming effect for each variant. Thus, any variant with an AIC value lower than that of other variants is said to better tame the uncertainty for the window period. Extending this logic further, any variant with a mean AIC value lower than other variants’ mean AIC value is said to better tame the uncertainty over the simulation run.

To determine if the mean of the AIC values of a recommender system framework is lower than those of other frameworks and statistically significant, we use the Wilcoxon signed-rank test, a nonparametric test to assess population means. The mean of the AIC values for the one-sided recommender system is lower but statistically insignificant than no recommender system framework (p-value = 0.224) for the student side. However, for the university side, the mean of AIC values for the one-sided recommender system framework is lower and statistically significant in comparison to no recommender system (p-value = 3.327e-09). Thus, hypothesis 1a, that the one-sided recommender system framework will better tame the irreducible uncertainty in comparison to no recommender system, is partially supported. For the student side, the mean of AIC values for the two-sided recommender system framework is lower and statistically significant in comparison to no recommender system (p-value = 3.045e-06) and the one-sided recommender system framework (p-value = 0.001). For university-side variants, the mean of the AIC values for the two-sided recommender system framework is lower and statistically significant in comparison to no recommender system (p-value = 3.327e-09) and the one-sided recommender system framework (p-value = 0.006). This result fully supports hypothesis 1b, that the two-sided recommender system framework will better tame the irreducible uncertainty in the system in comparison to no recommender and to the one-sided recommender system.

We repeated this analysis by varying the window size: 100, 200, 250, 350, and 400. We find that for all window sizes considered in the analysis, the mean of the AIC values of the two-sided recommender system is lower and statistically significant in comparison to the mean of the AIC values for no recommender system and the one-sided recommender system. We also see that the mean of the AIC values associated with the one-sided recommender system is lower and statistically significant for window sizes smaller than 300. This suggests that (a) the two-sided recommender system tames the irreducible uncertainty in the system when we consider shorter and longer duration (in comparison to no recommender system and to the one-sided recommender system), and (b) the one-sided recommender system is effective in taming the irreducible uncertainty in short spans (in comparison to no recommender system). We attribute this result to the two-sided recommender system framework’s ability to consistently learn the emergence and its effects on different sides of the platform. Consistently learning the effects of emergence in the platform enables the two-sided recommender system to better tame the uncertainty over both short and long periods.

## Improving Agent Performance

To quantify the effect on agent performance of taming uncertainty in the system, we perform multiple simulation runs by varying the values of the variables in our agent-based simulation model. For each simulation run, we select a variable and change its value from its baseline (the default values of the variables which represent high complexity are presented in Appendix C). Changing a variable’s value alters the complexity dynamics in the simulation model and simulates a gradual change in the irreducible uncertainty faced by agents. Table 7 lists the variables and their parameter values which were varied in the simulation runs. For each variable and its value, we run the simulation model 100 times where each run consists of 1000 periods. After one simulation run’s 1000 periods, the average fitness of students and universities is recorded and represents one data point. Each simulation run was computed for no recommender system, the one-sided recommender system, and the two-sided recommender system.

|  |  |
| --- | --- |
| **Variable** | **Values** |
| Types of Agents | Students and universities of Type 1 only, students and universities of Types 1 and 2, and students and universities of Types 1, 2, and 3 (baseline) |
| Social Network size | Maximum size of social network can be 5, 10 (baseline), or 15 |
| Incoming students each period | Maximum number of incoming students varied: 10, 50 (baseline), and 100 |
| Incoming universities each period | Maximum number of incoming universities varied: 5 (baseline) and 10 |
| Peer Grading | Maximum number of peers grading each assignment: 5 (baseline) and 10 |
| Discussion Board Participation | Maximum percentage of students in a course participating in discussion board: 50% and 90% (baseline) |

Table 7. Agent-based Simulation Setup for Hypotheses Testing

A linear regression model was estimated using the data to quantify the effect of each variable on the agents’ fitness while controlling for other variables. A linear regression model is appropriate to understand the effect of different solution mechanisms on agents’ performance in the CABS because a data point includes values of setup variables, the aggregate result, and a solution mechanism identifier for one simulation run with 1000 time periods. Therefore, the data for regression analysis is not from a time series, and different simulation runs are independent of each other. Equation 1 provides the linear regression model for average student fitness. The variable *AlgoType* is a categorical variable represented by no recommender system, the one-sided recommender system, and the two-sided recommender system. Table 8 provides the regression analysis output for the student side.

(1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Estimate** | **Std. Error** | **t value** | **Pr(>|t|)** |
| Intercept | 0.4217 | 1.408e-02 | 29.955 | <2e-16\*\*\* |
| One-sided Recommender System | 0.3115 | 2.174e-03 | 143.285 | <2e-16\*\*\* |
| Two-sided Recommender System | 0.6313 | 2.174e-03 | 290.417 | <2e-16\*\*\* |
| Agent Type | -0.08019 | 1.698e-03 | -47.212 | <2e-16\*\*\* |
| Social Network | 5.434e-04 | 3.112e-04 | 1.746 | 0.0808 |
| Incoming Students | 4.347e-05 | 3.698e-05 | 1.176 | 0.2398 |
| Incoming Universities | -6.908e-04 | 7.284e-04 | -0.948 | 0.343 |
| Peer Grades | 4.064e-04 | 7.284e-04 | 0.558 | 0.5769 |
| Discussion Board Participation | 6.766e-03 | 9.105e-03 | 0.743 | 0.4575 |
| Signif. Codes: ‘\*\*\*’ p<0.001, ‘\*\*’ p<0.01, ‘\*’ p<0.05  Adjusted R-squared: 0.9507  F-statistic: 1.085e+04 on 8 and 4491 DF, p-value: < 2.2e-16 | | | | |

Table 8. Regression Output for Student side

From the regression output in Table 8, we find that the one-sided recommender system increases the average student fitness by 0.31 when we control for other potential factors in comparison to no recommender system. This result partially supports hypothesis 2a. Also, the two-sided recommender system increases the average student fitness by 0.63 when we control for other potential factors in comparison to no recommender system. This result partially supports hypothesis 2b.

Similarly, equation 2 provides the linear regression model for average university fitness. From the regression output in Table 9, we find that the one-sided recommender system increases the average university fitness by 0.32 when we control for other potential factors in comparison to no recommender system. This result, along with student side result, supports hypothesis 2a. Also, the two-sided recommender system increases the average university fitness by 0.45 when controlling for other potential factors in comparison to no recommender system. To compare the one-sided recommender system with the two-sided recommender system, we rerun the analysis with the one-sided recommender system as the base. We find that the two-sided recommender system increases the average student (university) fitness by 0.31 (0.13) when we control for other factors in comparison to the one-sided recommender system. This result, along with the two-sided recommender system’s comparison with no recommender system, supports hypothesis 2b. From these results, we see that existing solution approaches can address traditional uncertainty. However, with increasing levels of uncertainty where we see irreducible uncertainty in the system, the traditional solution approach performs poorly. Table 10 summarizes the hypotheses, testing procedure and results.

(2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Estimate** | **Std. Error** | **t value** | **Pr(>|t|)** |
| Intercept | 0.0665683 | 0.0097822 | 6.805 | 1.14e-11 \*\*\* |
| One-sided Recommender System | 0.3210329 | 0.0015107 | 212.499 | < 2e-16 \*\*\* |
| Two-sided Recommender System | 0.4529699 | 0.0015107 | 299.832 | < 2e-16 \*\*\* |
| Agent Type | 0.0444447 | 0.0011803 | 37.654 | < 2e-16 \*\*\* |
| Social Network | -0.0006325 | 0.0002163 | -2.925 | 0.003467 \*\*\* |
| Incoming Students | 0.0017582 | 0.0000257 | 68.417 | < 2e-16 \*\*\* |
| Incoming Universities | -0.0127484 | 0.0005062 | -25.185 | < 2e-16 \*\*\* |
| Peer Grades | -0.0018196 | 0.0005062 | -3.595 | 0.000328 \*\*\* |
| Discussion Board Participation | 0.0046438 | 0.0063275 | 0.734 | 0.463043 |
| Signif. Codes: ‘\*\*\*’ p<0.001, ‘\*\*’ p<0.01, ‘\*’ p<0.05  Adjusted R-squared: 0.9574  F-statistic: 1.263e+04 on 8 and 4491 DF, p-value: < 2.2e-16 | | | | |

Table 9. Regression Output for University Side

|  |  |  |
| --- | --- | --- |
| **Hypotheses** | **Test Description** | **Result** |
| 1a (One-sided recommender system will better tame the complex search matching problem in comparison to no recommender system) | Mean AIC value of the one-sided recommender system framework is lower and (not) statistically significant than the mean AIC value of no recommender system for the student (university) side. | Partially Supported |
| 1b (Two-sided recommender system will better tame the complex search matching problem in comparison to both no recommender and a one-sided recommender system) | Mean AIC value of two-sided recommender system framework is lower and statistically significant than the mean AIC values of no recommender and one-sided recommender system framework for student and university side. | Supported |
| 2a (With increasing uncertainty on the digital platform, a one-sided recommender system will improve the average fitness of buyers and sellers in comparison to no recommender system) | A regression model is fitted to data generated by changing values of key variables. The one-sided recommender system increases the average student (university) fitness by 0.31 (0.32) when controlling for other potential factors in comparison to no recommender system. | Supported |
| 2b (With increasing uncertainty on the digital platform, a two-sided recommender system will improve the average fitness of buyers and sellers in comparison to both no recommender system and a one-sided recommender system) | A regression model is fitted to data generated by changing values of key variables. The two-sided recommender system increases the average student (university) fitness by 0.63 (0.45) in comparison to no recommender system and 0.31 (0.13) in comparison to the one-sided recommender system, when controlling for other factors. | Supported |

Table 10. Summary of Hypotheses Testing

In addition to the hypotheses, we find other interesting results from our analyses. First, from Table 8 we see that increasing the number of agent types in the system decreases the average student fitness by 0.08019. On the other hand, from Table 9 we see that increasing the number of agent types in the system increases the average university fitness by 0.044. While the absolute changes in fitness are small, increasing the number of agent types has a different effect on students and universities. We attribute this result to (a) an increase in competition, (b) an increase in uncertainty associated with agent types, and (c) different agent types having different dependency relationships with agents. Also, from Table 9, we find that increasing social network size decreases universities’ fitness by 0.00063, a change which can be attributed to increased variability in the signal. An increase in the incoming student rate increases universities’ fitness by 0.0017, a change which can be attributed to the completion of the objectives for universities. An increasing incoming rate of universities decreases universities’ fitness by 0.012, a loss which can be attributed to increased competition. Finally, an increase in the number of peers’ grading assignments decreases universities’ fitness by 0.0018, a shift which can be attributed to lower grades and subsequently lower course ratings.

# DISCUSSION AND CONCLUSION

Pervasive digitization has resulted in digital multi-sided platforms gaining significant importance across industries (Yoo et al. 2012). In part, this is due to their ability to facilitate value enhancing transactions (Parker et al. 2016), address mainstream and niche markets, introduce distributed and combinatorial innovation in the platform ecosystem (Yoo et al. 2012), and attract geographically distributed heterogeneous actors (Boudreau 2012). As firms find themselves operating in these platforms they face complex problems which cannot be addressed by the traditional strategy of collecting and processing data to reduce uncertainty (Tanriverdi et al. 2010). Instead, they need adaptive IS solutions which can enable agents to co-evolve and learn in the platform. The two-sided recommendation framework advanced in this paper provides such an adaptive IS solution which can tame complex problems in CABS. Empirical tests using our agent-based simulation model support the efficacy of the proposed two-sided recommender system to tame the complex search matching problem and improve agents’ fitness with increasing irreducible uncertainty in the CABS. We attribute this result to the two-sided recommender system’s complexity-informed adaptive design.

A one-sided recommender system framework has limitations in these environments specifically because it does not consider the impact and emergence on the other side. For example, if a new seller has entered the platform, buyers should see recommendations for the new seller. However, the one-sided recommender system, serving the buyer side, will recommend established sellers with high ratings, positive reviews, and competitive pricing. Even when a new seller offers competitive pricing, the one-sided recommender system may not recommend the new seller based on the buyer’s preferences towards sellers. In recommender system literature, this is often referred to as the “cold start problem.” In the case of a two-sided recommender system, the recommender considers the new seller’s participation objectives in addition to the buyers’ participation objectives. Therefore, the new seller has a higher probability of being recommended to buyers, at some expense to buyers’ preferences.

While the two-sided recommender system performed better to tame the complex search matching problem and improve agents’ fitness, the two-sided recommender system cannot fully reduce the irreducible uncertainty because agents develop smart responses to changes in the environment, and nonlinear interactions lead to unpredictable outcomes. The two-sided recommender framework needs to learn agents’ objectives, preferences, and constraints. Recommendations served based on some learned characteristics need constant revision because agents’ characteristics evolve over time. Also, to satisfy an objective, recommendations served to an agent may not have the intended result due to nonlinear interactions of other agents in the system. Thus, while adaptive moves by the two-sided recommender system tame irreducible uncertainty in the system, adaptive moves by other agents in the system do not fully reduce the uncertainty in the system.

## Limitations

Our work has several limitations and we highlight some of the important ones here, both for transparency as well as to motivate future work. Our results are limited to the market intermediary stream of platforms with a focus on the *platform provider* model of organizing platforms where the platform mediates participants’ interactions (Eisenmann 2008). Future studies may consider complex problems in other streams of platforms. The two-sided recommender system instantiated in this research is limited to the educational platform context. Generalizing the recommender system algorithm to other contexts may require adaptation of the recommender system’s objectives and behavioral rules. Also, the agent-based simulation model design incorporates abstract details of the educational platform context and excludes financial metrics. Future studies can consider detailed characteristics of the context, along with financial metrics and incorporate real data to simulate agent behavior.

A recent study by complexity researchers Komiyama et al. (2018) highlighted the biases inherent in machine learning algorithms which may manifest in unfair or suboptimal decisions for some agents on the platforms. In the context of two-sided recommendations made by platforms there may be a related concern of whether recommendations for a specific agent are influenced by broader platform objectives rather than just this single agent’s goals. This research does not consider any bias introduced by the two-sided recommender system or broader considerations of what fairness may mean in this ecosystem. The two-sided recommender system framework presented in this research can indeed be extended to include different objectives such as fairness of the recommendations served to agents. However how exactly this can be done is not obvious and needs detailed study.

Another limitation of this study is the limited set of design alternatives explored in our agent-based simulation. Specifically, we do not compare the efficacy of our two-sided recommender system with approaches such as search engines and price mechanisms. However, agents in our agent-based simulation can develop smart responses based on changing environments, mimicking the use of existing solution approaches. Finally, recommender systems are newer agents in the system which increase its complexity. Future studies may consider the trade-offs associated with the complexity and benefits introduced by an IT-based solution. Explicitly modeling how human agents in the system perceive these other agents may be a particularly promising line of inquiry, and it is likely that different types of perceptions influence the system’s emergence quite differently.

## Implications for Recommender Systems and Information Systems Research

Extant work in recommender systems literature has focused on the canonical tasks such as *find good items or predict an item’s relevance to a user* (Adomavicius and Tuzhilin 2005; Jannach and Adomavicius 2016). Current recommender system approaches serve a set of users and assume users belong to a single group (Godoy-Lorite et al. 2016). Also, existing evaluation procedures for recommender systems use hold-out samples from the original dataset (Herlocker et al. 2004), assuming that recommendations are served in a static system.

Drawing on the complexity science research, this paper conceptualizes the digital platform as a CABS where recommender systems serve agents that serve recommendations to other agents in the system. Complexity science informs us that in a CABS, agents develop nonlinear connections and adapt to the changing environment. Building on the properties of CABS, we discuss the importance of considering *complexity* in the search matching problem and its effect on the limited efficacy of the current one-sided recommendation framework. Also, complexity science suggests that an optimal solution to a complex problem does not exist (Tanriverdi et al. 2010) thereby highlighting the search for approaches to tame complexity. To tame the complex search matching problem, we presented the two-sided recommendation framework which considers emergence and its effects on the same side and other side of the platform.

In addition to multi-sided recommendations, this study has other important implications for recommender systems researchers. Evaluating these systems in the environment that they are deployed in and accounting for adaptive behavior on all sides may call for a different approach beyond testing by using hold-out data or even one-shot A/B testing experiments. Running forward-looking agent-based simulations may be an important component of how these systems should be evaluated.

Certainly, recommender systems research is an important subset of IS research, particularly in the last fifteen years. However, there are other implications as well. A significant direction of current IS research is the study of platform ecosystems, and our work here shows the importance (and value) of studying these through a complexity lens. In such cases we show in this paper how agent-based simulations of the complex system can help study important platform dynamics. Specifically, studying platforms from a design-perspective, as IS researchers are like to, can be significantly richer and more relevant by incorporating emergence and complexity dynamics.

## Implications for Complexity Science

The two-sided sided recommender system proposed in this research advances complexity science by designing an IT-based artifact (Hevner et al. 2004) that can augment the platform to tame the complex search matching problem. The two-sided recommender system augments human decision-making on the platform to achieve relatively better outcomes because the system considers the irreducible uncertainty inherent in the system and emphasizes consistent adaptability and learning. For complexity science, our results show that a two-sided recommender system provides significant benefit to tame the irreducible uncertainty in the system (on average, introduction of the two-sided recommender system improves students’ and universities’ fitness by 50% (buyers) and 87% (sellers), respectively, in comparison to the base case of no recommendation; such a significant improvement in agents’ fitness may lead to a virtuous cycle and provide substantial progress in taming irreducible uncertainty). While significant improvement in agents’ fitness is demonstrated in our experiments for the education domain, the practical implication of the absolute benefit achieved by introducing such an IT-based solution may be dependent on the context of future research.

The CABS property of connectedness informs us that an agent is not fully connected to other agents in the CABS (Page 2009; Tanriverdi et al. 2010). However, in the digital platform context considered in this paper, we note that the recommender system agent is connected to most (if not all) of the other agents in the CABS and influences a greater proportion of agents in comparison to other agents in the CABS. Despite this the recommender system’s intelligence cannot control other agents and assume the role of global controller because other agents may reject recommendations contextually, switch off access, or may develop different attitudes towards the recommender system.

This research contributes to complexity science by introducing an adaptive IT-based mechanism which can tame the complex search matching problem. While we show that multi-sided recommendations work for platforms such as the one studied here, it is quite likely that different types of complex systems need different methods for taming problems. There is little work in the literature on how to tame complex systems in the face of irreducible uncertainty and, by presenting one unique framework, our paper could lead to a renewed search for other interesting solutions to tame various other problems in platform ecosystems. Given the significant role that platform ecosystems currently play, this can be a particularly exciting direction for researchers in complexity science.

**Acknowledgements:** We would like to express our gratitude towards the Special Issue Senior Editors, Hüseyin Tanriverdi, Youngjin Yoo, and Bill McKelvey for their guidance and support. We would like to thank the workshop participants at the University of Texas at Austin and seminar participants at the University of South Florida for feedback and comments. We would like to thank Patricia Nickinson for proofreading this paper.

**REFERENCES**

Adams, S. 2013. “Are MOOCs Really A Failure?,” *Forbes*.

Adomavicius, G., and Kwon, Y. 2012. “Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques,” *IEEE Transactions on Knowledge and Data Engineering* (24:5), pp. 896–911.

Adomavicius, G., and Tuzhilin, A. 2005. “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions,” *IEEE Transactions on Knowledge and Data Engineering* (17:6), pp. 734–749.

Agrawal, A., Venkatraman, J., Leonard, S., and Paepcke, A. 2015. “YouEDU: Addressing Confusion in MOOC Discussion Forums by Recommending Instructional Video Clips,” *National Science Foundation*.

Allon, G., Bassamboo, A., and Cil, E. 2012. “Large-Scale Service Marketplaces: The Role of the Moderating Firm,” *Management Science* (58:10), pp. 1854–1872.

Amaral, L. A. N., and Uzzi, B. 2007. “Complex Systems - A New Paradigm for the Integrative Study of Management, Physical, and Technological Systems,” *Management Science* (53:7), pp. 1033–1035.

Ashlagi, I., Braverman, M., Kanoria, Y., and Shi, P. 2018. “Clearing Matching Markets Efficiently: Informative Signals and Match Recommendations,” Working Paper.

Asvanund, A., Clay, K., Krishnan, R., and Smith, M. 2004. “An Empirical Analysis of Network Externalities in Peer-to-Peer Music-Sharing Networks,” *Information Systems Research* (15:2), pp. 155–174.

Bindley, K. 2018. “Secrets of Better Uber and Lyft Rides,” *The Wall Street Journal*.

Boudreau, K. 2012. “Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation,” *Organization Science* (23:5), pp. 1409–1427.

Butler, B. S. 2001. “ Membership Size, Communication Activity, and Sustainability: A Resource-Based Model of Online Social Structures,” *Information Systems Research* (12:4), pp. 346–362.

Eisenmann, T. R. 2008. “Managing Proprietary and Shared Platforms,” *California Management Review* (50:4), pp. 31–53.

Epstein, J. M., and Axtell, R. L. 1996. *Growing Artificial Societies: Social Science from the Bottom Up*, MIT Press, Cambridge, MA.

Godoy-Lorite, A., Guimera, R., Moore, C., and Sales-Pardo, M. 2016. “Accurate and Scalable Social Recommendation Using Mixed-Membership Stochastic Block Models,” in *Proceedings of the National Academy of Sciences* (Vol. 113), pp. 14207–14212.

Hagiu, A. 2006. “Pricing and Commitment by Two-Sided Platforms,” *RAND Journal of Economics* (37:3), pp. 720–737.

Hagiu, A., and Yoffie, D. B. 2009. “What’s Your Google Strategy?,” *Harvard Business Review* (87:4), pp. 74–81.

Hardesty, L. 2010. “Explained: Linear and Nonlinear Systems,” *MIT News*. (https://news.mit.edu/2010/explained-linear-0226).

Herlocker, J., Konstan, J., Terveen, L., and Riedl, J. 2004. “Evaluating Collaborative Filtering Recommender Systems,” *ACM Transactions on Information Systems* (22:1), pp. 5–53.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. “Design Science in Information Systems Research,” *MIS Quarterly* (28:1), pp. 75–105.

Holland, J. 1998. *Emergence: From Chaos To Order*, Reading, MA: Addison-Wesley.

Hollands, F. M., and Devayani, T. 2014. “Why Do Institutions Offer MOOCs?,” *Online Learning* (18:3).

Jannach, D., and Adomavicius, G. 2016. “Recommendations with a Purpose,” in *Conference on Reccommender Systems*, Boston, MA, pp. 7–10.

Jones, Q., Ravid, G., and Rafaeli, S. 2004. “ Information Overload and the Message Dynamics of Online Interaction Spaces: A Theoretical Model and Empirical Exploration,” *Information Systems Research* (15:2), pp. 194–210.

Kanoria, Y., and Saban, D. 2017. “Facilitating the Search for Partners on Matching Platforms: Restricting Agent Actions,” Working Paper.

Komiyama, J., Takeda, A., Honda, J., and Shimao, H. 2018. “Nonconvex Optimization for Regression with Fairness Constraints,” in *International Conference on Machine Learning*, pp. 2742–2751.

Konnikova, M. 2014. “Will MOOCs Be Flukes?,” *The New Yorker*.

Lee, C., Venkatraman, N., Tanriverdi, H., and Iyer, B. 2010. “Complementarity-Based Hypercompetition in the Software Industry: Theory and Empirical Test, 1990-2002,” *Strategic Management Journal* (31:13), pp. 1431–1456.

Li, Y., and Li, H. 2017. “MOOC-FRS: A New Fusion Recommender System for MOOCs,” in *2nd Advanced Information Technology, Electronic and Automation Control Conference*.

McKelvey, B., Tanriverdi, H., and Yoo, Y. 2015. “Complexity and Information Systems Research in the Emerging Digital World,” MIS Quarterly.

Merali, Y., Papadopoulos, T., and Nadkarni, T. 2012. “Information Systems Strategy: Past, Present, Future?,” *The Journal of Strategic Information Systems* (21:2), pp. 125–153.

Nan, N. 2011. “Capturing Bottom-up Information Technology Use Processes: A Complex Adaptive Systems Model,” *MIS Quarterly* (35:2), pp. 505–532.

Nan, N., and Tanriverdi, H. 2017. “Unifying the Role of IT in Hyperturbulence and Competitive Advantage via a Multilevel Perspective of IS Strategy,” *MIS Quarterly* (41:3), pp. 937–958.

Page, S. E. 2009. *Understanding Complexity*, Chantilly, VA: Teaching Company.

Pappano, L. 2012. “The Year of the MOOC,” *The New York Times*.

Parameswaran, A., Venetis, P., and Garcia-Molina, H. 2011. “Recommendation Systems with Complex Constraints: A Course Recommendation Perspective,” *ACM Transactions on Information Systems* (29:4), pp. 1–33.

Parker, G. G., Van Alstyne, M. W., and Choudary, S. P. 2016. *Platform Revolution*, W. W. Norton & Company.

Petrongolo, B., and Pissarides, C. A. 2001. “Looking into the Black Box: A Survey of the Matching Function,” *Journal of Economic Literature* (39:2), pp. 390–431.

Pizzato, L., Rej, T., Chung, T., Koprinska, I., and Kay, J. 2010. “RECON: A Reciprocal Recommender for Online Dating,” *4th ACM Conference on Recommender Systems (RecSys)*, Barcelona, Spain, pp. 207–214.

Prabhakar, S., Spanakis, G., and Zaiane, O. 2017. “Reciprocal Recommender System for Learners in Massive Open Online Courses (MOOCs),” *16th International Conference on Web-Based Learning*, Cape Town.

Romanyuk, G. 2016. “Ignorance Is Strength: Improving the Performance of Matching Markets by Limiting Information,” Working Paper.

Roth, A. 1982. “The Economics of Matching: Stability and Incentives,” *Mathematics of Operations Research* (7:4), pp. 617–628.

Shani, G., and Gunawardana, A. 2011. “Evaluating Recommendation Systems,” in *Recommender Systems Handbook*, Springer US, pp. 257–297.

Shapiro, C., and Varian, H. R. 1999. *Information Rules: A Strategic Guide to the Network Economy*, Boston, Massachusetts: Harvard Business School Press.

Solon, O. 2018. “Not Just Homes Any More: Airbnb Expands into Hotels and Luxury Spots,” *The Guardian*, San Francisco.

Tanriverdi, H., and Lim, S. 2017. “How to Survive and Thrive in Complex, Hypercompetitive, and Disruptive Ecosystems? The Roles of IS-Enabled Capabilities,” in *International Conference on Information Systems*.

Tanriverdi, H., Rai, A., and Venkatraman, N. 2010. “Reframing the Dominant Quests of Information Systems Strategy Research for Complex Adaptive Business Systems,” *Information Systems Research* (21:4), pp. 822–834.

Thomas, L. D. W., Autio, E., and Gann, D. M. 2014. “Architectural Leverage: Putting Platforms in Context,” *The Academy of Management Perspectives* (28:2), pp. 198–219.

Tiwana, A. 2013. *Platform Ecosystems*, Morgan Kaufmann.

Ubell, R. 2017. “How the Pioneers of the MOOC Got It Wrong,” *IEEE Spectrum*.

Wiggins, R. R., and Ruefli, T. W. 2002. “Sustained Competitive Advantage: Temporal Dynamics and the Incidence and Persistence of Superior Economic Performance,” *Organization Science* (13:1), pp. 81–105.

Wiggins, R. R., and Ruefli, T. W. 2005. “Schumpeter’s Ghost: Is Hypercompetition Making the Best of Times Shorter?,” *Strategic Management Journal* (26:10), pp. 887–911.

Xia, P., Liu, B., Sun, Y., and Chen, C. 2015. “Reciprocal Recommendation System for Online Dating,” *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Paris, France, pp. 234–241.

Yang, D., Piergallini, M., Howley, I., and Rose, C. 2014. “Forum Thread Recommendation for Massive Open Online Courses,” *7th International Conference on Educational Data Mining*, London.

Yoo, Y., Boland Jr., R. J., Lyytinen, K., and Majchrzak, A. 2012. “Organizing for Innovation in the Digitized World,” *Organization Science* (23:5), pp. 1398–1408.

Zhenghao, C., Alcorn, B., Christensen, G., Eriksson, N., Koller, D., and Emanuel, E. 2015. “Who’s Benefiting from MOOCs, and Why,” *Harvard Business Review* (22).

**Appendix A – Mapping CABS Concepts and Digital Platform**

|  |  |  |
| --- | --- | --- |
| **Complex Adaptive Business System (CABS) Concepts and Properties** | **Description** | **Digital Platform** |
| **Agent** | Individual actors or basic entities of actions | Agents in the CABS of the digital platform (buyers, sellers, platform, technological components of the platform, and platform owner(s)). Individual users (consumers/buyers and vendors/sellers) that participate in transactions over the platform. Other information technology (IT) based agents include recommender systems, payment systems, and application programming interfaces (APIs). Platform owner(s) define architecture and governance mechanisms of the platform. |
| *Attributes* | Internal states of agents | Internal states of buyers such as preferences (products and service category), cognitive and interaction styles, cognitive and monetary capability, constraints, participation objectives (low cost, product variety, customization); internal states of sellers such as product and service offerings, pricing policy, inventory constraints, participation objectives (market access, diversified consumer base, enhanced reputation); internal state of technological components such as parameters, datasets, algorithms, models, and objectives; internal state of the owner(s) such as principles, objectives, and management styles. |
| Diversity of agents | Although agents have different internal states, they share certain characteristics such as type of transactions and interactions (buy, sell, rent, messages, reviews), participation objectives, and constraints (transaction and interactions channels). Simultaneous differences and similarities between these agents retain a diverse set of agents that are neither too similar nor too dissimilar. Diversity is primarily limited to consumers and sellers. |
| *Behavioral rule* | Schemata that govern attributes and behaviors of agents | Mental activities and models in the form of decision rules, heuristics, algorithms, and functions, which allow an agent to process its knowledge and perform intelligent actions to complete objectives. |
| Adaptation of agents | Agents learn from their prior transactions and experiences, emerging trends on the platform, and actions of connected peers, and they evaluate efficacy of rules. Learning provides input to behavioral rules, identifies new rules, and evaluates existing rules of the agent. However, an agent cannot assimilate knowledge to perfectly adapt itself to the environment because other agents adapt in response. |
| **Interaction** | Mutually adaptive behaviors | Symbiotic associations between agents are developed because of participation in transactions and interactions. |
| *Connection* | Relational links among agents (Connectedness) | Links between agents form and evolve over time. Buyers develop relational links with other buyers through messages and discussions; with sellers through ratings and transactions; with the platform through a search engine, the recommender system, and browsing. Sellers develop relational links with the platform through a search engine, the recommender system, the transaction database, and browsing. Platform owner(s) have relational links to platform components as they define the architecture and governance mechanisms of the platform. |
| Interdependence | A focal buyer develops asymmetric dependency relationships with other buyers for product reviews and experiences, with sellers for products or services and with platform components (search or recommendation). These dependencies may change across transactions; a focal seller develops asymmetric dependency relationships with other sellers for pricing and features and with platform components (search, recommendation, database). As platform dynamics evolve, the platform develops asymmetric dependencies on certain consumer and seller segments; thus, performance outcome for an agent is a function of its actions and other agents’ actions in the system |
| *Flow* | Movements of resources | The platform and its owner(s) enable diverse agents to participate in transactions with movement of ideas, information, knowledge, and experiences. |
| **Environment** | Medium for agents to operate on and interact with | The digital platform offers an easily accessible medium for buyers and sellers to share, interact, and transact with other buyers and sellers. The platform also offers a medium to interact through search engines, recommender systems, and databases. |
| *Structure* | Topography of an environment and its relationship to agents | The platform owner(s) set the architecture and governance mechanisms for the digital platform and its components. However, these rules do not *control* actions of buyers and sellers; they allow generative agent actions and behavior. |

Table 11. Mapping between CABS Concepts and Digital Platform (adapted from Nan (2011) and Tanriverdi et al. (2010))

**Appendix B – Conceptual Definition of CABS Model**

|  |  |  |
| --- | --- | --- |
| **Element** | **Conceptual Definition** | **Description** |
| **Student** | Individual human actors seeking registrations, feedback on submissions, and completion of courses | Students are individual entities that seek to satisfy their participation objective in their area(s) of interest. |
| *Individual Differences* | Objectives, learning rate, fitness, areas of interest, constraints, grades, propensity of accepting recommendation | Students differ in their objectives (audit, certification, or specialization), learning capability, area(s) of interest, overall experience (intrinsic) on the platform (fitness), constraints in the form of number of active courses, discussion participation, and differing views on accepting recommendations. |
| *Mental activities* | Learning from platform | Students learn about new offerings from the platforms using browse and search functionalities. Students also learn about universities and their suitability with course descriptions, reviews, and ratings. |
| Learning from recommender system | The recommender system provides actionable recommendations (courses, discussion boards, peers, tutorials) based on the focal student’s transaction history, learned objectives, and changes on the platform. |
| Social learning | Students learn and share information and knowledge about course offerings, dependencies, best practices, platform properties, and recommender system experiences with other students on the platform via messages, reviews, ratings, or forums. |
| **University** | Institutional human actors that offer courses and provide feedback | Universities offer courses in their area(s) of interest for students on the platform; the courses can be audited or taken for certification or within the specialization. |
| *Individual Differences* | Objectives, fitness, course rating, utilization goal, area(s) of interest, constraints, propensity to accept recommendation | Universities differ in their objectives; type of their offerings (audit, certification, or specialization); utilization goals; maximum number of active offerings, number of engaged students, and views on recommendations of the platform. |
| *Mental activities* | Learning from platform | Universities learn about their own offerings (through ratings and reviews) and existing offerings of other universities and their characteristics on the platform. |
| Learning from recommender system | Universities learn new action possibilities with recommendations (courses to offer, intervention on discussion board). |
| Social learning | Universities learn about experiences and opportunities from other connected universities. |
| **Educational Platform** | The digital platform that provides features and functionality for students and universities to participate in transactions | The platform provides functionality to universities to offer and manage courses and to students to register and complete courses. The platform has collaboration tools such as messaging, reviews, and ratings. |
| **Recommender system** | Recommender system on the platform | Each agent (student and university) receives recommendations on the next set of actions. These recommendations take different forms suitable for different agents based on their internal states. Recommendations offered by the system form inputs to the behavioral rules of a focal agent. They may introduce new action possibilities to be considered or reinforce existing action possibilities. |
| *Mental activities* | Learning from the platform | The recommender system learns offerings, prior transactions, patterns, preferences, objectives, and attributes. |
| Learning from Students | The recommender system learns students’ search and browsing patterns, interactions, transactions, grades, discussion participation, and propensity of accepting recommendations. |
| Learning from Universities | The recommender system learns universities’ offerings, competition, utilization goals, and propensity of accepting recommendations. |
| **Human actor-platform interactions** | Mutually adaptive behaviors between human actors (students and universities) and platform | The platform simultaneously enables and constrains actions of human actors on the platform. |
| *Actor-platform links* | Direct use of platform | The platform features allow users to undertake a set of actions such as offering courses, registering, submitting assignments, grading, rating and reviewing courses, paying for certification and specialization, participating in discussion boards, grading, and messaging. |
| *Movement of tangible and intangible resources* | Knowledge acquired through experience, information provided by the platform | The platform provides raw and aggregate information to its users in the form of reviews, ratings, completion rates, and trends. Over time, users learn the importance of different information sources and associate different weights to develop input to their behavioral rules. |
| **Human actor-recommender system interactions** | Mutually adaptive behaviors between human actors (students and universities) and recommender system | By serving recommendations (input to behavioral rules), the recommender system influences human actor behavior to improve the actor’s experience and objectives on the platform. Acceptance or rejection of recommendations updates the internal state of the recommender system and updates the input for the recommender system’s behavioral rule. |
| *Actor-recommender system links* | Direct use of recommender system | Actors receive recommendations from the platform. Actors retain the right to accept or reject any or all recommendations served by the recommender system. |
| *Movement of tangible and intangible resources* | Knowledge acquired through experience, information provided by the recommender system | Actors receive from the recommender system actionable recommendations that are accepted or rejected. Over time, actors develop knowledge about the benefits, drawbacks, and constraints of the recommender system while the recommender system learns usage patterns of the focal actor. |
| **Interpersonal interactions** | Mutual-influence behaviors among students | Students on an educational platform are often geographically distributed. They use platform features like reviews, messages, ratings, and peer evaluations as ways to influence other students’ behavior. |
| Mutual-influence behaviors among universities | A focal university may influence other universities by sharing experiences and information about student trends, best practices, and offerings. |
| *Interpersonal ties* | Interpersonal relationships between students | Students develop personal networks with other students. These networks are developed over time as a focal student interacts with other students from a course, area(s) of interest, forums, assignments and projects. |
| Interpersonal relationships between universities | Universities share information with other universities through personal ties or official channels. |
| Interpersonal relationships between students and universities | Students develop relationships with universities by following their offerings and activities on the platform. Similarly, universities develop relationships with student groups. |
| *Movement of intangible resources* | Information transfer between students | Personal networks allow flow of information across students. Information may be in the form of experiences, preferences, activities, and/or assignment evaluations. |
| Information transfer between universities | Information flows across universities in the form of experiences, best practices, preferences, and objectives. |
| Information transfer between students and universities | Students and universities exchange information via social channels such as reviews, ratings, and forums. Similarly, universities provide feedback on students’ submissions. |
| **Environment** | Educational platform | An Internet-based multi-sided education platform attracts a diverse set of students and universities that participate in transactions to fulfil varying objectives; the platform benefits by extracting profits from transactions. |
| *Social and operational structure* | Rules governing platform transactions | Human agents are required to follow the platform’s architecture, governance mechanisms, and rules set forth by the platform owner(s). These rules identify the framework for transactions and conduct on the platform. |

Table 12. Conceptual Definition of the CABS Model (adapted from Nan (2011))

**APPENDIX C – Definition and Possible Values of Agent Attributes**

|  |  |  |
| --- | --- | --- |
| **Student** | | |
| **Attribute** | **Definition** | **Possible values** |
| ID | Unique identifier for each student | Integer |
| Fitness | The student’s ability to achieve her objectives. The value is assigned (uniformly random) when she enters the system. A student with low fitness has a higher probability of exiting the system before her objectives are complete. | Unit Interval (0, 1] |
| Active | Each student is marked active when she enters the system. | Time when the student is active on the platform |
| Inactive | A student is marked inactive when she exits the system due to completion of her objectives or low fitness. | Time when the student becomes inactive |
| Exit period | Each student is assigned a random number to exit the platform. | Uniformly random between [0, 75] |
| Learning rate | Cognitive capability of a student. The value is assigned (uniformly random) when she enters the system. | Unit Interval (0, 1] |
| Probability of accepting recommendation | A student’s propensity to accept a recommendation. Value is assigned (random with mean 0.7 and standard deviation of 0.3) when she enters the system. | Unit Interval (0, 1] |
| Probability of Comments | A student’s propensity to provide good feedback to her peers. | Unit Interval (0, 1] |
| Areas of interest | Before each student enters the system, we randomly draw and select two areas of interest. | Two values from {1, 2, 3} |
| Type | Assigned (uniformly random) when she enters the system. | Value from {1, 2, 3} |
| Grades | For each course, a focal student receives grades for assignments, peer grading, and discussion boards. | Key value pair (course: grade) |
| Specialization | If the student is of type 3, she identifies a university and pursues its specialization. | Specialization offered by a university |
| Completed courses | Time when a course is completed. | Key value pair {course ID: time of completion} |
| Active courses | Time when a student registers for a course. | Key value pair {course ID: time of registration} |

|  |  |  |
| --- | --- | --- |
| **University** | | |
| **Attribute** | **Definition** | **Possible values** |
| ID | Unique identifier for each university | Integer |
| Fitness | A university’s ability to achieve its objectives. Assigned (uniformly random) when it enters the system. A university with low fitness has a higher probability of exiting the system before its objectives are complete. | Unit Interval (0, 1] |
| Active | Each university is marked active when it enters the system. | Time when the student is active on the platform |
| Inactive | A university is marked inactive when it exits the system due to low fitness. | Time when the student becomes inactive |
| Exit Period | Each university is assigned a random number to exit the platform. | Uniformly random between [0, 75] |
| Probability of accepting recommendation | A university’s propensity to accept a recommendation. The value is assigned (uniformly random) when it enters the system. | Unit Interval (0, 1] |
| Areas of Interest | Randomly assigned two areas of interest. | Two values from {1, 2, 3} |
| Type | Assigned (uniformly random) when it enters the system. | Value from {1, 2, 3} |
| Courses | Set of offered courses. | Set of course IDs |
| Specialization | If the university is of type 3, define a specialization with course structure (suggested prerequisites). | Specialization ID with course structure |
| Utilization Goal | Assigned (uniformly random) when it enters the system; determines the utilization rate threshold for each university to improve fitness. | Unit Interval (0, 1] |

|  |  |  |
| --- | --- | --- |
| **Course** | | |
| **Attribute** | **Definition** | **Possible values** |
| ID | Unique identifier for each course. | Integer |
| Seller | ID of university which is offering the focal course. | Integer |
| Start Time | Time when the course will start (assigned randomly). | Time + Random Integer |
| End Time | Time when the course will end (assigned randomly). | Start\_Time + Random Integer |
| Number of Assignments | Number of assignments in the course (End\_Time – Start\_Time). | Integer |
| Cohort | Set of students who have registered for the course. | Set of students |
| Times Offered | Number of times the course has been offered. | Integer |
| Area of Interest | Assigned (uniformly random) from offering university’s areas of interest. | A value from {1, 2, 3} |
| Difficulty Level | Each area of interest has courses with increasing level of difficulty (5 levels). Each course is assigned (uniformly random) a difficulty level. | A value from {1, 2, 3, 4, 5} |
| Type | Type of course (audit, certification, or specialization) determined by offering university’s type. | A value from {audit, certification, or specialization} |
| Rating | Average rating of the course. | A value from {1, 2, 3, 4, 5} |

**APPENDIX D – Summary of Agent-based Model**

|  |  |  |
| --- | --- | --- |
| **Element** | **Conceptual Definition** | **Description** |
| **Student (Buyer)** | Students seeking registration, completion of courses, feedback and grades | Students browse and register for courses, submit assignments, peer grade, and participate in discussion boards. |
| *Individual differences* | Learning rate, fitness, constraints, grades, propensity of accepting recommendation | Randomly assigned initial values for these variables differentiate each student. Students seek to complete their objectives and improve fitness; changes in values of these variables are based on the focal student’s actions and experiences on the platform. |
| *Mental activities* | Learning from platform | Students learn about existing and new course offerings, ratings, and popularity. |
| Learning from recommender system | Recommendations received from the system alter the inputs to behavioral rules for the focal student. |
| Social learning | Students receive inputs from their connections and enter only if they receive positive feedback. After each period, the focal student receives messages about fitness from other students in her social network which influence focal student’s fitness. |
| **University (Seller)** | Universities offering courses | Universities offer courses on the platform that students audit or take for certification or specialization. |
| *Individual differences* | Fitness, ratings, utilization goal, constraints, accepting recommendation | Randomly assigned initial values to attributes differentiate each university. |
| *Mental activities* | Learning from platform | Universities learn about rival offerings and their characteristics on the platform. This input allows the university to determine changes, if any, to its offerings. |
| Learning from recommender system | Universities learn new action possibilities and/or reinforce existing possibilities under consideration such as course offerings and discussion boards. |
| Social learning | After each period, the focal university receives messages about fitness from other universities in its social network which influence the focal university’s fitness. |
| **Recommender system** | Recommender system on the platform | The recommender system serves recommendations to one or multiple sides, based on its framework. |
| *Individual differences* | Set of recommendations offered | The recommender system creates different internal states for different agents (students and universities) based on past transactions, trends, and agent actions. Recommendations served to a focal agent at any time are different from any other previous time. |
| *Mental activities* | Recommender system’s adaptability to emergence on platform | As agents (students and universities) participate in transactions, the recommender system recalibrates its internal state based on new input. Each internal state of the recommender system uses different input for decision rules based on the internal state of the agent, historic transactions, agent objectives, and platform objectives. |
| **Human actor-platform interactions** | Mutually adaptive behaviors between human actors (students and universities) and platform | The platform simultaneously enables and constrains actions of human actors on the platform. This enables the platform to facilitate value enhancing transactions for agents from different sides. |
| *Actor-platform links* | Direct use of platform | All human actors can access information on the platform and participate in transactions. |
| *Movement of tangible and intangible resources* | Knowledge acquired through experience and use, information provided by the platform | The platform provides raw and aggregate information to its users in the form of reviews, ratings, completion rates, and trends. |
| **Human actor-recommender system interactions** | Mutually adaptive behaviors between human actors (students and universities) and recommender system | Human actors execute behavioral rules to learn from the recommender system and the recommender system executes its behavioral rules to learn from human actors. |
| *Actor-recommender system links* | Direct use of recommender system | Human actors receive recommendations on the platform. |
| *Movement of tangible and intangible resources* | Knowledge acquired through experience and use, information provided by the recommender system | The recommender system learns students’ and universities’ propensity towards recommendation acceptance; students and universities learn new action possibilities. |
| **Interpersonal interactions** | Mutual-influence behaviors among students | Students execute behavioral rules for social learning via interpersonal ties. Learning may take different forms such as platform and recommender systems experience, best practices, and peer grades. |
| Mutual-influence behaviors among universities | Universities execute behavioral rules for social learning via interpersonal ties. Learning may take different forms such as platform and recommender systems experience, offerings, and utilization patterns. |
| Mutual-influence behaviors among students and universities | Students influence universities by ratings, reviews, and preferences; universities influence students by changing their offerings and feedback. |
| *Interpersonal ties* | Interpersonal relationships among students | Students develop and evolve their social networks. |
| Interpersonal relationships among universities | Universities develop and evolve their social networks. |
| Interpersonal relationships among students and universities | Students follow popular courses and area(s) of interests; universities follow student groups that are interested in their area(s) of specialization. |
| *Movement of tangible and intangible resources* | Information transfer among students | Students share information pertaining to their fitness and course ratings through interpersonal interactions. |
| Information transfer among universities | Universities share information pertaining to their offerings and fitness through interpersonal interactions. |
| Information transfer among students and universities | Changes in students’ preferences shift demand to different courses; changes in universities’ offerings shifts students’ preferences in area(s) of interests. |
| **Environment** | Educational platform | Online courses are accessible to students on the platform. |
| *Social and operational structure* | Rules governing platform transactions | The digital platform provides process channels to facilitate registration, course offering, grading, interaction, message passing, and discussion boards. |

Table 13. Summary of Agent-Based Model Design

**APPENDIX E – Pseudo-code and Description of the Agent-Based Simulation**

For each period *t* {

Add students to pool of potential participants randomly varied around expectation of existing active population

Add universities to pool of potential participants randomly varied with expectation

For each agent (student or university) from potential population {

Randomly draw an area of interest

If draws equal threshold {

Randomly develop Social Network of focal agent (student or university)

If average fitness of social network members exceeds randomly generated unit interval {

Activate focal student with randomly generated values

}

Activate university with randomly generated values

}

}

For each active university, offer courses {

If threshold of maximum active courses not exceeded {

If recommendation accepted {

**If one-sided recommender system {Offer courses in areas of interest with type and increasing difficulty}**

**If two-sided recommender system {Offer courses in an area of interest with type from high demand and increasing difficulty}**

}

Else {

Offer courses from areas of interest with type

}

Randomly determine start and end time, and number of assignments for offered courses

Reoffer courses that have completed

}

}

For each active student, register for courses {

If maximum active courses not exceeded {

If recommendation accepted {

**If one-sided recommendation {randomly register for eligible and increasing difficulty courses}**

**If two-sided recommendation {randomly register for eligible, increasing difficulty, and focus area of interest}**

}

Else {randomly register for eligible courses}

}

}

For each active student, submit assignments {

For each ongoing course {

If all prerequisites are completed {increment learning rate of focal student}

Determine grade for submission normally distributed on learning rate

}

}

For each active student, receive peer grades {

For each ongoing course {

If all prerequisites are completed {increment learning rate of focal student}

Determine part I of grade for submission normally distributed on learning rate

**If recommendation accepted for either recommender system {Add peers from course cohort with similar areas of interest and relational links}**

Else {Add random peers from the course cohort}

Determine part II of grade for submission by averaging peers normally distributed comments

}

}

For each active student, receive discussion board feedback {

For each ongoing course {

If all prerequisites are completed {increment learning rate of focal student}

If university representative participates {increment learning rate of focal student}

Determine part I of grade for submission normally distributed on learning rate

**If recommendation accepted for either recommender system {Add peers from course cohort with similar areas of interest and relational links}**

Else {Add random peers from the course cohort}

Determine part II of grade for discussion board by averaging peers normally distributed comments

}

}

For each active student { //Course concludes at the end of *t*

For each ongoing course, rate courses on completion [End Time == *t*] {

Determine average grade for assignments, peer assignments, and discussion boards

If (grade<0.2) {rating=1}

Else If (grade >= 0.2 and grade < 0.4) {rating=2}

Else If (grade >= 0.4 and grade < 0.6) {rating=3}

Else If (grade >= 0.6 and grade < 0.8) {rating=4}

Else If (grade >= 0.8) {rating=5}

Determine focal student’s expectation

If expectation exceeds grade {course rating = average rating and 5 stars}

Else If expectation meets grade {course rating = average rating and 3 stars}

Else {course rating = rating}

}

}

Update social network of students and universities with recent relational links

For each active Student, update fitness {

SNFitness = Determine average weighted fitness of students from social network

AvgGrade = Determine average grade across courses

If (SNFitness >= randomly generated unit interval) {increment fitness}

Else {decrement fitness}

If (AvgGrade >= randomly generated unit interval) {increment fitness}

Else {decrement fitness}

If (Objective completing) {increment fitness}

Else {decrement fitness}

}

For each active University, update fitness {

AvgUtilization = Determine average utilization across all offered courses

AvgRating = Determine average rating across all offered courses

SNFitness = Determine average fitness of universities from social network

If (AvgUtilization >= randomly generated unit interval) {increment fitness}

Else {decrement fitness}

If (AvgRating >= randomly generated unit interval) {increment fitness}

Else {decrement fitness}

If (SNFitness >= randomly generated unit interval) {increment fitness}

Else {decrement fitness}

}

For each active Student, update beliefs {

Determine completed areas of interest, specializations

If (all areas of interest are completed) {exit platform}

If (fitness < randomly generated unit interval) {exit platform}

If (fitness >= randomly generated unit interval) {increment student type}

If (randomly assigned exit time) {exit platform}

}

For each active University, update beliefs {

If (fitness >= randomly generated unit interval) {add new area of interest}

Else {exit platform}

If (randomly assigned exit time) {exit platform}

}

}

For each period *t*, students and universities are added to a pool of potential participants. This step simulates the potential participant pool that forages information to determine whether the platform provides value. A higher population signals the popularity of the platform and attracts more potential participants. For each potential participating student, we randomly determine a social network of active students in the system. If the average fitness of students in the social network is greater than or equal to a randomly generated unit interval, we activate the focal student from the potential pool; otherwise, we inactivate the student (she exits from the platform). Each activated student’s internal state is instantiated with values such as the probability of accepting recommendations, objectives, areas of interest, type, fitness, and data structures to track the focal agent’s activities over time. In the case of universities, we add participants in each period to the potential pool as we assume that the number of active universities does not influence potential participants. Each university is also associated with ten other active universities. This represents connections of agents as they enter the complex system. Each activated university’s internal state is instantiated with values such as probability of recommendations, areas of interest, maximum courses, type, fitness, and data structures to track the focal university’s activities over time.

Active universities offer courses in their area(s) of interest that can be audited or taken for a certification, or are part of a specialization, based on the type of focal university. Each university can offer a specific number of courses randomly determined during its activation. Courses have an associated difficulty level (one to five). A course with a difficulty level of one is introductory, while a course with a difficulty level of five is advanced in its area of interest. Because courses of the same difficulty level in a given area of interest are offered by different universities, the study simulates competition, as seen in actual internet-based educational platforms. Active students register for eligible courses which are the set of all courses from their area(s) of interest, not completed or registered, and type of the focal student. Selection of courses from this set is random and every active university’s courses are considered.

Each activated course has a start and end time that is randomly determined. At the time of offering, each course is also initialized with individual assignments, peer assignments, and discussion boards. For example, if a course start time and end time are 3 and 8 (duration of 5 periods), respectively, each registered student will submit 5 individual assignments and 5 assignments to be graded by peers in the course and contribute to 5 discussion board questions. In the individual assignment, each registered student’s learning rate determines her grade. If the focal student has completed all lower-difficulty courses in the area of interest, her learning rate is randomly incremented before the grade is assigned. The grade is randomly determined from normal distribution with the learning rate as the mean. In the case of peer-graded assignments, other students from the course grade a focal student’s assignment. There are two equally weighted parts to this grade: (a) the focal student’s efficacy (learning rate), which determines the quality of the submission, and (b) a peer’s propensity to provide detailed feedback on the submission. For the first part, the grade is determined based on the logic presented for the individual assignment. However, the value is scaled between 0 and 0.5. For part two, peers are randomly chosen from the course to provide a grade. Each student is assigned a uniformly random value for “probability of comments” at the time of activation. This represents the probability of providing detailed feedback to peers. Using this value as the mean, the normally distributed random value is determined for each of the selected peers. The average value of these peers’ feedback determines part two of the grade. For discussion boards, universities (via assumed faculty) randomly participate in discussion boards. Assuming an enhanced learning experience for involved students, the learning rate is randomly incremented. Similar to the peer grading assignment, peers interact in discussions and receive grades. Finally, for each student, grades are aggregated at the end of each course as an average across all the assignments. Students rate completed courses (on a scale of 1 to 5) based on their expectation (a random number between students’ learning rate and 1) and actual grades. The focal student rates a course highly if the final grades exceed her expectation, and logs a low rating if grades are significantly lower than expected.

Finally, students and universities update their fitness based on their experiences during time *t.* Students leave the platform if their objectives are complete or their fitness is less than a uniformly random number from unit interval. Similarly, universities leave the platform if their fitness is less than a uniformly random number from the unit interval. In real world MOOC platforms, for reasons beyond the control of students and universities, students and universities leave the platform even when their experiences on the platform are positive. We replicate this phenomenon by randomly forcing agents to exit the platform.

**Author Biographies**

**Onkar Malgonde** is an Assistant Professor at the Operations Management & Information Systems, College of Business, Northern Illinois University. He received a B. Engg in Information Technology in 2010 from University of Pune and a PhD in MIS from the University of South Florida in 2018. His research interests include software engineering, machine learning, and digital platforms. His research has been published in Empirical Software Engineering, Electronic Markets, Workshop on Information Systems and Technology (WITS), International Conference on Design Science Research in Information Systems and Technology (DESRIST), and Americas Conference of Information Systems (AMCIS).

**He Zhang** is an assistant professor in the Information Systems and Decision Sciences department in the Muma College of Business at the University of South Florida. His research interests include healthcare information management, Big Data, and production and inventory management. Zhang's research has been published in journals including Mathematical Programming, Decision Support Systems and ACM Transactions on Management Information Systems.

**Balaji Padmanabhan** is the Anderson Professor of Global Management, Professor in the Information Systems Decision Sciences Department and Director of the Center for Analytics & Creativity. He received a B.Tech in Computer Science in 1993 from Indian Institute of Technology (IIT) Madras and a PhD from the Stern School of Business New York University (NYU) in 1999. His interests and expertise include analytics and business Intelligence, designing analytics algorithms for business applications, managing analytics, building and evaluating predictive models, patterns discovery in data, business value of analytics, enabling citizen data science and applications of analytics in churn, healthcare, recommender systems, fraud detection and elections. His research has been published in the premier computer science and business journals and conferences. He serves on the editorial board and program committees of many leading academic journals and conferences in the field.

**Moez Limayem**, the dean of the USF Muma College of Business, has enhanced the college’s profile and increased its resources by building program demand and strengthening relationships with stakeholders, alumni, and business leaders. He was appointed dean 2012, coming from the Sam M. Walton College of Business at the University of Arkansas. Since his start, the business college has received five multi-million dollar naming gifts and its programs have risen in national rankings. The recipient of numerous professional awards, Limayem’s research has focused on the intersection of technology with the consumer, academic and business worlds. His most recent publication is "Building an Informing Business School: A Case Study of USF's Muma College of Business." His articles have appeared in top journals such as MIS Quarterly, Management Science and Information Systems Research. He received an MBA and a PhD in business administration from the University of Minnesota and has taught at Laval University in Canada, City University of Hong Kong and Lausanne University in Switzerland.

1. Emergence is the gradual creation of new patterns and trends which distort the existing order in the CABS (Holland 1998). Agents’ preferences and constraints gradually evolve as objectives are completed and changes in the environment introduce new objectives which in turn impact other agents. [↑](#footnote-ref-2)
2. These approaches primarily serve the buyers’ side of the platform. On the sellers’ side, platforms typically provide aggregate data and reporting systems. [↑](#footnote-ref-3)
3. https://lytics.stanford.edu/ [↑](#footnote-ref-4)
4. The initial assignment of fitness value presented here differs from prior agent-based simulation model approaches where fitness values are computed for each period based on the focal agent’s attributes (Nan 2011; Nan and Tanriverdi 2017). These studies consider a CABS where the number of agents is constant. In our agent-based simulation model, the number of agents (students and universities) changes every period. [↑](#footnote-ref-5)
5. Artificially imposing the complexity dynamics would entail specifying the types of connections, actions, behaviors, and responses for each agent. However, artificially imposed dynamics will give rise to predictable behaviors among agents and a near-equilibrium state of the overall system. [↑](#footnote-ref-6)