

©Copyright 2022

Emisa Nategh

Learning-based Models for Intelligent Decision Systems with  
Application to Crowdsourced Systems and Healthcare

Emisa Nategh

A dissertation  
submitted in partial fulfillment of the  
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

Michael R Wagner, Chair

Mark Hillier

Yong-Pin Zhou

Program Authorized to Offer Degree:  
Foster School of Business

University of Washington

**Abstract**

Learning-based Models for Intelligent Decision Systems with Application to Crowdsourced Systems and Healthcare

Emisa Nategh

Chair of the Supervisory Committee:  
Michael R Wagner  
Information Systems and Operations Management

Technologies like artificial intelligence, the Internet of Things, quantum computing, robotic process automation, and genomics have ushered in the fourth industrial revolution. Digital (crowdsourcing) platforms are one of the technologies driving the fourth industrial revolution due to their new capabilities. Rather than relying on connections forged in the physical world, digital crowdsourcing platforms enable businesses to connect with users online and gather data that helps them improve their service offerings.

A digital crowdsourcing platform is a network that expedites valuable connections, interactions, and exchanges between people. These exchanges include sharing information, selling products, and offering services. Seven of the top ten most valuable companies in the world are now based on a platform business model, and according to a McKinsey research report, more than 30% of global economic activity could be mediated by digital crowdsourcing platforms by 2025 (Schenker [2019]). Well-known crowdsourcing platforms like Uber, Airbnb, Amazon, StackOverflow, YouTube, Facebook, and GrubHub are so successful because they save users time and money, while also eliminating traditional industry brokerages and providing potential for people to profit from their skills.

Therefore, given the positive economic and environmental impacts of crowdsourcing platforms, it is important to study how their pricing policy works. Our business partner in this study, Convoy, is a digital freight network transporting goods around the world with endless capacity (number of carriers in the network) and zero waste (eradicating empty miles that happens when trucks do not have any cargo in the back) by eliminating carbon emissions from our planet. In other words, Convoy operates a digital marketplace for shippers and carriers, and it is like Uber for trucking. Convoy uses digital crowdsourcing technology to make freight more efficient, reduce shippers' costs, and increase carriers' earnings. Convoy uses technology to solve waste and inefficiency problems in the \$800B trucking industry, which generates more than 87M metric tons of wasted CO<sub>2</sub>-equivalent emissions from empty trucks/miles (Convoy [2022]). As more carriers join the Convoy's network, capacity increases and shippers see lower prices per mile and higher quality. A common theme in crowdsourcing platforms including Convoy is that the platform has a limited budget and a set of jobs (shipments in our case) to accomplish by a pool of online workers (drivers on Convoy app). A key point to making these crowdsourcing platforms efficient is to design proper incentive pricing policy for workers. The tradeoff between efficiency and workers' incentives makes the pricing decisions complex in these platforms. In a realistic crowdsourcing setting, each worker has specific preferences which define the set of jobs the worker willing to do. For instance, based on driver's interest in shipment millage, location, time, number of stops, and price the driver can only accomplish some shipments, and not all. Note that the platform has a limited budget and should focus on picking the right driver and offering the right price. The focus of our study is on designing pricing mechanisms (dynamic and auction) for empty trucks. In this thesis, we design pricing mechanisms for the following setting: there is a digital platform who has a set of jobs (shipments) and a limited budget. To do the shipments, there is a pool of drivers. Each driver has certain interests which makes the driver eligible to do only certain shipments, and not all. Moreover, each driver

has a (private) minimum cost (unknown to the driver), which is the amount the driver is willing to gain for accomplishing a shipment. We assume that a driver can do only one job (one or multiple shipments).

Dynamic pricing is a pricing strategy based on real-time market demand and supply data that applies variable prices instead of a fixed price. There are often uncertainties during pricing decision making, such as unknown demand. Firms aim to optimize their revenue while learning demand. Therefore, another critical aspect of pricing is understanding customer decision-making patterns. One of the undesirable effects of dynamic pricing is that it can lead to a type of customer behavior that is referred to as a strategic behavior (due to human rationality) in the revenue management literature (Tang and Netessine [2009]). Strategic customers decide not only which products to purchase but also when to purchase them (at which price). Therefore, a potential drawback of dynamic pricing is that strategic customers contribute to delay in sales.

Demand uncertainty in dynamic pricing causes an inherent trade-off between exploration and exploitation. The exploration involved in learning demand and suboptimal prices and the exploitation which is the price that appears to be the best price so far occurs due to cost minimization. In general, exploitation occurs due to revenue maximization. One way of modeling this exploration-exploitation trade-off is using a classic reinforcement learning framework called a multi-armed bandit modeling problem. Multi-armed bandit (MAB) problems are powerful frameworks for algorithms to make sequential decisions under uncertainty with an exploration-exploitation trade-off and are well-known models in reinforcement and online learning. There are three essential variants of the MAB problems: stochastic bandits, adversarial bandits, and Markovian bandits. Many algorithms have been proposed to solve these three types of MAB problems. Our focus in Chapter 2 is on the Upper

Confidence Bound (UCB) algorithm, which is a deterministic algorithm that focuses on exploration-exploitation based on a confidence boundary. There are different variants of the UCB algorithms, such as UCB1, UCB2, KL-UCB, lil' UCB, etc. In Chapter 2, we choose the UCB1 algorithm, which selects the best arm based on an optimistic estimate and achieves regret that increases only logarithmically with the number of actions taken. Regret in MAB problems is defined as the expected loss in each time step due to not playing the optimal arm. The objective of our study in this chapter is to propose an optimal incentive (UCB-based) dynamic pricing mechanism to reduce an imbalance problem in a trailer-sharing system subject to a budget limitation.

There is a basic pricing mechanism known as the posted price that a firm chooses a price and offers to customers, take-it-or-leave-it. This mechanism is transactionally efficient (because the posted price is the platform's best and final offered price and there is no negotiation), easy to communicate, and can be executed quickly. However, it burdens the firm with setting the right price based on customers' willingness-to-pay and the firm's supply. There are other trends in pricing mechanisms, such as an auction and haggling, in which the firm allows customers to decide on the price by submitting their bids (negotiating the price). Although these pricing mechanisms are transactionally more burdensome, they simplify willingness-to-pay discovery and potentially increase revenue. Wang [1993] analyzes when an auction pricing mechanism is a better selling mechanism than posted price mechanism. He demonstrates the auction selling revenue dominates a posted price when the characteristics of the distribution of the bidder's value are heterogeneous. For example, Mukhopadhyay and Kekre [2002] demonstrate the use of an auction pricing mechanism to find the lowest-cost supplier all over the world (see more in Brown et al. [2010]).

A well-designed auction pricing mechanism causes the final price to reflect the willingness-

to-pay or true preference. With modern technology and algorithms, an auction pricing mechanism can be implemented quickly and is superior at generating revenue in a digital environment relative to a posted price mechanism (Cachon [2020]). Our objective in Chapter 3 is to design an optimal incentive auction pricing mechanism to maximize the cumulative profit via empty trailer return in a trailer-sharing system in the presence of rational drivers. We also compare the performance of our auction pricing mechanism with our dynamic posted price mechanism, which is measured in terms of “regret.”

Finally, we study using natural language processing (NLP) and machine learning in health-care. The use case that interests us is the application of a specific form of NLP, namely Named Entity Recognition (NER), to automate data extraction from clinical records. New tools are being developed to speed up manual extraction by using NER, which, when properly trained, allows a model to automatically identify and extract relevant data from a body of text. In this study, we evaluate the effectiveness of using Amazon Comprehend Medical (ACM) in assessing a cohort of patients with multiple myeloma seen at the Seattle Cancer Care Alliance (SCCA). This tool uses NER to extract relevant data from unstructured texts and compile them into datasets. More specifically, it recognizes a feature in a document, extracts it, and classifies it as a drug, symptom, diagnosis, or another appropriate classification. It also records the location of the feature in the clinical records. We compared the NER features to manually extracted data to explore how the two sources complement each other and to evaluate the accuracy of NLP algorithm in ACM.

We focus on ACM applications for multiple myeloma for this study. Multiple myeloma is a form of cancer that affects bone marrow plasma cells and currently has no known cure. In the U.S., there were around 35,000 new cases in 2021 with a 5-year relative survival rate of 55% between 2011 and 2017, making it the second most common type of blood cancer

and 14<sup>th</sup> most common cancer overall. Researchers continue to investigate the most effective ways to treat multiple myeloma, and current treatments can limit the spread of the disease and sometimes cause complete remission. We chose this disease because of the substantial data we have able to collect as a result of the patients' relatively long survival time, the regular monthly follow-ups, and the SCCA's robust data due to being one of the leading bone marrow transplant centers in the country.

**Keywords:** Amazon Comprehend Medical Service, Auction pricing, Crowdsourced platform, Dynamic pricing, Machine learning, Multi-armed bandit, Multiple Myeloma, Natural language processing, UCB algorithm

## TABLE OF CONTENTS

	Page
List of Tables . . . . .	iii
List of Figures . . . . .	iv
Chapter 1: Introduction . . . . .	1
1.0.1 The Stochastic Multi-Armed Bandit Problem . . . . .	5
1.0.2 Stochastic Bandit Policies . . . . .	7
1.0.3 Budget Limited: Bandit Problems with Pulling Cost . . . . .	8
1.0.4 Thesis Outline and Research Contributions . . . . .	10
Chapter 2: Dynamic Pricing Mechanism for Trailer-sharing . . . . .	17
2.1 Introduction . . . . .	17
2.2 Literature Review . . . . .	23
2.2.1 Truckload Brokerage Services . . . . .	23
2.2.2 Incentivizing Workers in Crowdsourcing . . . . .	24
2.2.3 Dynamic Pricing Model . . . . .	25
2.2.4 Dynamic Fleet Repositioning and Balancing Problem . . . . .	25
2.2.5 Multi-armed Bandit Problems . . . . .	26
2.2.6 Regret Minimization . . . . .	28
2.3 Problem Setting and Model . . . . .	28
2.3.1 Problem Setting for Dynamic Pricing Mechanism . . . . .	28
2.3.2 Dynamic Pricing Mechanism . . . . .	32
2.3.3 Problem Setting for Facility Selection Algorithm . . . . .	34
2.3.4 Facility Selection Algorithm . . . . .	35
2.4 Results . . . . .	36
2.5 Conclusions and Managerial Insights . . . . .	39
2.6 Future Work . . . . .	40

Chapter 3:	Auction Pricing Mechanism for Trailer-Sharing . . . . .	41
3.1	Introduction . . . . .	41
3.2	Literature Review . . . . .	45
3.3	Problem Setting and Algorithm . . . . .	47
3.4	Results . . . . .	49
3.4.1	Proof of Truthfulness . . . . .	49
3.4.2	Proof of Individual Rationality . . . . .	51
3.4.3	Proof of Computational Efficiency . . . . .	52
3.5	Conclusions and Future Work . . . . .	56
Chapter 4:	Performance Evaluation of clinical entities extraction: Amazon Com- prehend Medical versus Manual Data Extraction . . . . .	57
4.1	Introduction . . . . .	57
4.1.1	Patients Information . . . . .	59
4.1.2	Natural language processing . . . . .	62
4.2	Methods . . . . .	64
4.2.1	Bidirectional Long Short-Term Memory . . . . .	64
4.2.2	Data Analysis . . . . .	65
4.2.3	Quantitative Analysis Techniques . . . . .	68
4.2.4	Statistical Analysis . . . . .	69
4.2.5	Descriptive Analysis . . . . .	70
4.2.6	Outliers . . . . .	72
4.2.7	Qualitative Analysis Techniques . . . . .	76
4.3	Result . . . . .	78
4.3.1	Accuracy of Extraction through Amazon Comprehend Medical Ser- vice . . . . .	78
4.4	Discussion . . . . .	81
4.4.1	Feature Context . . . . .	82
4.4.2	Inconsistencies in Clinical Notes . . . . .	84
4.4.3	Challenges of Using NLP in Oncology . . . . .	85
4.5	Conclusions . . . . .	85
Chapter 5:	Conclusions . . . . .	87
Bibliography	. . . . .	89

## LIST OF TABLES

Table Number	Page
4.1 Summary of Patients Characteristics . . . . .	60
4.2 Summary of Datasets . . . . .	62
4.3 Summary for the Natural Language Processed Dataset . . . . .	63
4.4 Accuracy of an assortment of features from the automatically extracted data when compared to the manual dataset . . . . .	79
4.5 Characteristics of the manually and automatically constructed datasets . . .	79

## LIST OF FIGURES

Figure Number	Page
2.1 Booking a shipment job [Convoy-Team, 2020] . . . . .	19
2.2 A shipment job [Hubert, 2019] . . . . .	20
2.3 GoPlus: ETR flexibility . . . . .	21
2.4 Relation between driver’s desired location and relocation request . . . . .	21
2.5 Optimal offline price $p^*$ . . . . .	31
2.7 Facility Selection . . . . .	35
2.8 Mean Rate of Choosing Optimal Price in 500 Simulation . . . . .	37
2.11 Utility for uniform cost distribution in $[0.1, 0.9]$ , varying budget. . . . .	38
2.9 Price Convergence . . . . .	38
2.10 Average Regret Convergence . . . . .	38
2.12 Average regret of the pricing mechanisms with increasing budget. . . . .	39
3.1 Computational Efficiency . . . . .	53
3.2 Total Reward vs. Number of arms and Budget . . . . .	54
3.3 Individual Rationality of Auction Pricing Mechanism . . . . .	55
3.4 Comparing Dynamic and Auction Pricing Mechanisms Performance . . . . .	56
4.1 Data Extraction . . . . .	60
4.2 Number of entities in different datasets extracted through NLP . . . . .	64
4.3 Distribution Information from Manual Extraction . . . . .	66
4.4 Distribution Information from NLP . . . . .	67
4.5 Distribution of Infections . . . . .	69
4.6 Feature distribution for infected patients . . . . .	70
4.7 Predicting TTNT results . . . . .	71
4.8 Regression model coefficients visualization . . . . .	71
4.9 Dataset scatterplots by the dependent variable . . . . .	73
4.10 Performance of Local Outlier Factor Removal Method . . . . .	74
4.11 Performance of Brute-Force Outlier Removal Method . . . . .	75
4.12 Performance of Mark Outlier Values Outlier Removal Method . . . . .	75

4.13 Performance of Exacerbate Outliers Removal Method . . . . .	76
4.14 Frequency of top 10 terms in the first year of treatment . . . . .	77
4.15 Extraction accuracy in myeloma therapy (MT) and medication (M) datasets	80

## ACKNOWLEDGMENTS

Sincere appreciation goes first to my advisor Prof. Michael Wagner for his guidance and mentorship throughout my Ph.D. His knowledge in various topics and critical thinking have greatly inspired and shaped me as a researcher. He did a great job pushing, prodding, questioning, interpreting, and suggesting as the research progressed. From his incredible intuition and deep understanding of problems, I learned how to formulate and approach research problems. His unreserved support and trust have enabled me to explore various ideas. I was very fortunate to have Michael as my thesis advisor and to work with and learn from him for the last three years. I would also like to thank my mentor Dr. David Coffey in the healthcare area research. Working with him on natural language processing, which became the last chapter of this thesis, was exciting and rewarding. I have learned immensely from David, and from his insights and mentorship. I am deeply thankful to the committee members, Prof. Mark Hillier, Prof. Yong-Pin Zhou, and Prof. Archis Ghate, for their willingness to serve on the committee and their valuable feedback.

Special thanks to Prof. Sham Kakade and Prof. Kevin Jamieson from the Computer Science department for exposing me to the fundamentals of bandit algorithms and reinforcement learning, from which my thesis work started. This research benefited from discussion with Omid Sadeghi. It was also a rewarding experience to collaborate with Convoy Inc. and Fred Hutchinson Cancer Research Center as a research scientist scholar.

Finally, I would like to thank Michael Wagner once again, who served as an unwavering source of support and encouragement from the early to the later days.

## DEDICATION

To my Mom (Hamideh), Neda, and Mohsen



## Chapter 1

### INTRODUCTION

Brabham [2013] posits the following definition: “Crowdsourcing is an online, distributed problem-solving and production model that leverages the collective intelligence of online communities to serve specific organizational goals”. According to the World Bank *www.worldbank.org* and industry research by *www.crowdsourcing.org*, crowdsourcing is expected to be a \$300 billion industry by 2025. With the growth of the Internet over the past few years, a wide range of organizations have turned to an online labor market, such as clickworker, TaskRabbit, and Fiverr, all of which include a pool of workers available online. Online crowdsourcing platforms, such as Innocentive, Openideo, Amazon Mechanical Turk, uTest, Upwork Inc., 99Designs, and Cad Crowd, become popular nowadays. IT has the capability to create multi-sided platforms, which connect diverse users throughout the world. These platforms ease transactions and interactions in a wide range of contexts, including, transportation, e-commerce, product review sites, classified ads, hospitality, and dating. Crowdsourcing is the IT-mediated engagement of crowds for task completion and improves efficiency, while decreasing transaction costs, dependence on crowds, and monetary exchange. In a crowdsourcing market, requesters post jobs using either a smartphone app or an Internet webpage, and workers decide whether to accept or complete the task, often in exchange for money. Workers are incentivized monetarily or non-monetarily (e.g., extra time off and work flexibility) to respond to a variety of tasks. Workers have some limitations on the number of jobs they can perform, and the requester has a fixed budget for paying workers. Given the limited budget constraints of these platforms, the ultimate goal of a requester is to design a payment scheme or assign jobs to workers in a way that maximizes the profit or minimizes the costs. Topics around platforms and crowds have become focal points of research in in-

formation technology (IT), computer science, marketing, and operations management. Yet, there is little research available on crowdsourcing.

Crowd-based models leverage the digital platforms' capability to connect buyer(s) and heterogeneous sellers for various purposes. In other words, Convoy operates a digital marketplace for shippers and carriers, and it is like Uber for trucking. Convoy uses digital crowdsourcing technology to make freight more efficient, reduce shippers' costs, and increase carriers' earnings. Therefore, given the positive economic and environmental impacts of crowdsourcing platforms, it's important to study how their pricing policy works. Convoy uses technology to solve waste and inefficiency problems in the \$800B trucking industry, which generates more than 87M metric tons of wasted CO<sub>2</sub>-equivalent emissions from empty trucks/miles (Convoy [2022]).

A pricing mechanism is one of the key components in any revenue model. According to an article in Harvard Business Review (Mohammed [2012]), pricing goods and services is the fastest way for a business to profit. And according to Monroe [1973], the price of goods and services sends a message to customers. Anglin et al. [2003] has shown that setting the price too high (or low) affects the marketability of goods or services. Furthermore, the seller's initial choice of list price is crucial in the marketing of the goods (or services). A too high initial price may discourage purchasing by potential customers and risk of having the goods on the market for a long time. Conversely, the too low initial price, could effect a rapid sale at the cost of selling the goods for less than could have been received with better market exposure. Also, according to a study by McKinsey (Mohammed [2012]), researchers found that with just a 1% increase in price, assuming demand remains constant, profit would increase on average by 11%. Therefore, obtaining the right pricing strategy is of great importance. Nowadays, businesses from start-ups to large companies apply revenue management optimization tools to derive optimal price (Özer and Phillips [2012], Tang et al.

[2013]). Hence, the focus of our study in Chapters 2 and 3 of this thesis is on designing pricing mechanisms (dynamic and auction) for empty trucks.

In recent years, the topic of dynamic pricing is one of the emerging areas of research and has attracted significant attention from researchers in operations management, computer science, economics, and marketing. Dynamic pricing is an effective operational research tool implemented in various industries, including online on-demand platforms. It is a pricing strategy based on real-time market supply and demand data that applies variable prices instead of a fixed price. There are often some uncertainties during pricing decision making, such as unknown demand. Another critical aspect of pricing in revenue management is understanding customer decision-making patterns. Strategic customers decide not only which products to purchase but also when to purchase them (at which price). According to an article in *The Economist* (Schumpeter [2016]), the practice of dynamic pricing goes back to the 1980s, when American Airlines varied flight tickets' prices to compete with discounters such as People's Express. As a result, the usage of dynamic pricing was expanded to the other airlines, hotels, railways, and car-rentals. With the emergence of e-commerce, dynamic pricing has become an effortless and simple way to track users' decisions and, as a result, monitor demand and adjust prices dynamically with minor costs. One of the undesirable effects of dynamic pricing is that it can lead to a type of customer behavior that is referred to as a strategic behavior (due to human rationality) in revenue management literature (Tang and Netessine [2009]). According to a recent report by the U.S. Commerce Department (of Commerce. [2022]) the total e-commerce sales for 2021, were estimated at \$218.5B which experienced a growth rate of 14.2% ( $\pm 0.9\%$ ) compared to the previous year. Amazon, one of the global ecommerce giants, makes up a portion of these online sales by frequently (2.5 million times a day Mehta et al. [2018]) adjusting prices. Also, industry research at 360pi, a price analytics company, indicates retailers such as Best Buy and Walmart change prices of their products about 50,000 times per month. Amazon Relay, Walmart

Freight Handler, and Uber Freight are considered Convoy’s competitors. Therefore, similar to Amazon, Walmart, and Uber, we consider studying a dynamic pricing system for Convoy in Chapter 2.

Chapter 2 of this thesis is dedicated to studying a data-driven approach. There are conflicting goals when seeking the two objectives of revenue maximization and demand learning, known as the exploration-exploitation trade-off. In the dynamic pricing problems, demand uncertainty causes an inherent trade-off between exploration (learning demand and suboptimal prices) and exploitation (the price that appears to be the best price so far). The seller would like to use exploration, i.e., sequential trials of alternative price offerings, so as to learn about the market responses and hopefully to find out the optimal price offering. An exploration policy is defined as the set of decision rules, where this policy at time  $t$  is non-anticipative and can only be based on all past price offering decision and the prior knowledge of the demand function. In general, exploitation occurs due to revenue maximization. Therefore, initially, the algorithm is more focused on exploration. It is a discovery phase with the goal being to accurately estimate the parameters of the price demand relation. Gradually, the algorithm switches to the exploitation phase, where it focuses more on revenue maximization rather than exploration of the parameters. One way of modeling this exploration-exploitation trade-off is using a classic reinforcement learning problem called a multi-armed bandit (MAB) <sup>1</sup> modeling problem. A bandit problem is a sequential game between a player and an oracle. The multi-armed bandit problem is where players are given a slot machine with  $K$  arms (bandits) with each arm having its own probability distribution of success. The original motivation of MAB models dates back to 1933

---

<sup>1</sup>One application of MaB is exactly in A/B testing. In general, A/B testing aims for learning the impact of all variations with statistical confidence, and only makes business decisions after that; whereas MAB optimizes the conversions (or other primary goal) during the testing phase. In A/B Testing, we focus on answering “Is the difference between advertisement A and advertisement B statistically significant?” On the other hand, in MAB algorithm, we focus on answering “Which, out of multiple possible advertisements, is the best option?” By combining the explore and exploit stages, MAB offers lower costs since the optimization starts while data is still being collected.

when William R. Thompson introduced a bandit problem for clinical trials. In this bandit problem, different treatments were available for a certain disease, and a physician would decide which treatment to prescribe for the next patient. Multi-armed bandit problems are powerful frameworks for algorithms to make sequential decisions under uncertainty with an exploration-exploitation trade-off and are well-known models in online learning. The MAB modeling approach has a long history dating back to the early 1950s (Robbins [1952]). Over the past 20 years, an enormous amount of scholarship in MAB has been completed in multidisciplinary areas. It has received widespread attention in operations management literature, and many algorithms have been proposed to solve MAB problems (Gittins [1989], Gittins et al. [2011], Auer et al. [2002a], Garivier and Cappé [2011], Kaufmann et al. [2012a,b], Thompson [1933], Bubeck and Cesa-Bianchi [2012], Agrawal and Goyal [2013]). MAB problems arise and play an important role in numerous industrial domains, such as recommendation systems in movies, hotels, and restaurants; ad placement on webpages; auction market design (bid optimization through auction adjustment over time); price assignment on crowdsourcing platforms; and other applications. One of the most well-known examples is dynamic pricing in online environments, when a seller is selling an item through an online system. When a new customer arrives, the seller picks a price to offer to this customer. The customer makes a take-it-or-leave-it decision. The seller’s goal is to maximize the total profit.

### *1.0.1 The Stochastic Multi-Armed Bandit Problem*

In a finite-horizon ( $T$ ) bandit problem, at each time step  $t$ , the player chooses an action  $A^t$  from a given set of actions  $\mathcal{A}$  based on the history of choices and their rewards; then, the oracle reveals a reward  $R_t$ . The reward for each arm is drawn independently from a fixed distribution that depends on the chosen arm but not on the time step  $t$  and it is not known to the algorithm. The basic MAB algorithm observes the reward of the chosen arm at each time step—but not for the other arms. Therefore, the algorithm needs to try out different

arms to gather new information (exploration). The performance of MAB algorithms is measured in terms of “regret” (Berry and Fristedt [1985]), which is defined as the expected loss in each time step due to not playing the optimal arm. The objective of the player is to choose actions that lead to the smallest possible cumulative regret over all  $T$  rounds (In our setting ). In a  $K$ -armed bandit problem, the player selects one of the  $K$  possible actions ( $\mathcal{A} = \{a_1, a_2, \dots, a_K\}$ ) known as arms with unknown reward distribution to maximize the cumulative reward or minimize the cumulative regret (which is the difference between the cumulative reward of the oracle that knows the true expected rewards and that of the player) over  $T$  rounds. In the original formalization (Robbins [1952]), each arm  $a_i, i = 1, \dots, K$  corresponds to an unknown probability distribution  $q_{a_i}$ , and rewards  $R_t^{a_i}$  are independent draws from the distribution  $q_{a_i}$  corresponding to the chosen arm. For  $a_i, i = 1, \dots, K$ ,  $\theta_{a_i}$  denotes the mean of  $q_{a_i}$  (mean reward of arm  $a_i$ ). Let

$$\theta^* = \max_{a_i, i=1, \dots, K} \theta_{a_i}$$

and

$$a^* = \arg \max_{a_i, i=1, \dots, K} \theta_{a_i}$$

The algorithm’s cumulative reward with the best-arm benchmark equals  $\theta^* \cdot T$ , which is the expected reward of always playing an optimal arm. Therefore, in the stochastic setting, it is easy to see that the regret (over  $T$  total rounds) can be written as

$$\text{Reg}_T = T \cdot \theta^* - \sum_{t=1}^T \mathbb{E}[\theta_{A^t}].$$

The regret is always non-negative for every bandit. As we mentioned above, the goal of an MAB algorithm is typically to minimize  $\text{Reg}_T$  (equivalent to maximize reward  $\sum_{t=1}^T \mathbb{E}[\theta_{A^t}]$ ).

### 1.0.2 Stochastic Bandit Policies

As we mentioned above, in the stochastic multi-armed bandit problem, the player is facing the exploration-exploitation dilemma (the challenge of making decisions under uncertainty). Researchers have proposed a variety of approaches that tackle this dilemma from different perspectives. Among these aspects, the greedy algorithm is the simplest policy in any MAB setting, which at each time step, it chooses the arm with the current highest expected reward (Tse and Leung [2022], Bayati et al. [2020], Kumar et al. [2015], Qin et al. [2015]). However, this algorithm represents low efficiency in most bandit problems. Because the player performs insufficient exploration. To take exploration into account, a number of greedy algorithm variants have been introduced. One of the simplest approach is  $\epsilon$ -greedy algorithm, that the player follows the greedy policy. It means the algorithm pulls the arm with the highest estimate with probability  $(1 - \epsilon)$ , and it selects a random arm with probability  $\epsilon$ .  $\epsilon$  (the exploration parameter) guarantees that all possible arm is continuously pulled over the time. Therefore,  $\epsilon$ -greedy fails to achieve asymptotic convergence to the optimal behavior. Because it is desirable to stop exploration process once the optimal arm is learnt.

$\epsilon$ -first is another variant of the greedy algorithm (Miao and Chao [2021], Raharjo [2016], Lacerda et al. [2015]). This algorithm splits the exploration phase from exploitation phase, explicitly. By assuming the time horizon is  $T$ , the player (uniformly) samples an arm to pull for the first  $\epsilon T$  time steps (exploration phase) and then selects greedily for the remaining  $(1 - \epsilon)T$  time steps. The first  $\epsilon B$  of the total budget  $B$  is used for exploration, and the remaining  $(1 - \epsilon)B$  for exploitation. This algorithm might wrongly choose a suboptimal arm to pull within the exploration phase. Therefore, similar to  $\epsilon$ -greedy algorithm, this algorithm does not converge to the optimal behavior.

Upper Confidence Bound (UCB)-based algorithms, efficiently tackle this problem. It is proven that UCB1 achieves asymptotically optimal performance (efficient finite-time) re-

gret bounds, which only differ from the best optimal bound by a constant factor (Xia and Yap [2018], Liu et al. [2014], DaCosta et al. [2008], Kleinberg [2004b]). They also proved that in simulations this algorithm is outperformed compared to the budget-limited  $\epsilon$ -first.

### *1.0.3 Budget Limited: Bandit Problems with Pulling Cost*

The standard MAB model does not consider pulling costs. There are some real-world applications such as costly medical treatments or the shortest driving path scenario that we need to take pulling costs into account. To date, most bandit models with such cost constraints focus on the cases when the arm pulling is costly only within the exploration phase. However, in many real-world applications, the exploitation phase is also costly, that is limited by a cost budget (e.g. online advertising). In the budget-limited MAB problems, in which pulling an arm is costly in both the exploration and exploitation phases, it is also limited by a budget. As a result, in this thesis we address one of the variants of the classic bandit model, the budget-limited bandit model, in which pulling an arm (taking actions) is costly (and each action might incur a different cost), and is limited by a fixed budget. Perhaps the easiest way is to think about a robot with limited battery, and each of its action would consume some amount of energy, therefore depleting the battery. Assuming the robot wants to maximise its total payoff until the battery is completely depleted, it can use one of the budget-limited MAB algorithms. In the auction domain, one can think about an online advertisement company who received an order from a client, which says that they want to maximise the number of clicks/impressions out of \$5K. The company can use this money to buy online advert slots. Therefore, in this case the \$5K is the budget and placing bids to buy slots are the actions.

The budget-limited MAB problems should operate over a finite time interval, because the overall budget is finite. Therefore, it is crucial to guarantee that for any budget size, the regret is bounded. This means, the performance of the designed policy has to be efficient

so that it is always close to the optimal arm. Also, the arm pulling policy should be able to learn the optimal arm in the long run. In order to achieve the aforementioned goal, we need a policy that converges to the optimal policy with probability 1 as the budget tends into infinity (there is enough budget to learn the optimal behavior). Auer et al. [2002b] designed UCB1 policy which achieves bounded regret in finite time as well as having optimal asymptotic convergence. UCB1 pulls each arm once at the beginning, then at each subsequent time step  $t$ , it chooses arm  $i$  that maximizes

$$\hat{\theta}_i^t + \sqrt{\frac{2 \cdot \ln(t)}{n_i^t}}$$

where  $\hat{\theta}_i^t$  is the estimate of arm  $i$ 's expected reward value, and  $n_i^t$  is the number of times the algorithm pulled arm  $i$  until time step  $t$ . Other algorithms in the class of upper confidence bound typically show significant shortfalls (fail to achieve good performance) compared to the UCB1 (Esterle [2018], Saito et al. [2015], Huang et al. [2014], Garivier and Cappé [2011]). An overview among all UCB-based pulling policies indicates the following properties in UCB1 perform better: 1. computational cost, 2. experimental performance, 3. asymptotic convergence, and 4. regret bound. Other MAB algorithms provide either significantly worse theoretical guarantees, or have poor performance in practice, since they might result in sequence of pulls in which the total pulling cost exceeds the budget limit (Zimmert and Seldin [2021], Trovo et al. [2018], Abraham et al. [2013]).

This budget can be chosen by the player (company). If there is no loss function, then it is trivial (the larger the budget the better). However, we can consider some kind of extra loss function say  $L(B)$  which depends on the set budget. The total payoff is then equals collected rewards minus  $L(B)$ . Again, think about a robot which can carry different sizes of batteries. Bigger batteries provide longer lifespan but also come with heavier weight. Therefore, each action's cost will also depend on the budget size (the heavier the robot, the more difficult to move the whole body). Therefore, there is definitely a question of trading

off between budget sizes. Note that within the budget-limited algorithms which we focus on in this thesis, the set of arms that we can pull with respect to the remaining budget decreases over time as the pulling budget decreases.

In our research work, our payment is not allowed to exceed a certain limit of expense that we charged shippers in our annual contract. Therefore, our both dynamic (Chapter 2) and auction pricing (Chapter 3) mechanisms require a different optimal behavior. Given this, the main objective of this thesis is to design pricing mechanisms that efficiently tackle the budget-limited bandit problem. The budget constraint also helps the company publish next jobs (hauling empty trailers from surplus facilities to deficit facilities) for balancing facilities more wisely. For instance, the company can think about creating combined jobs including one shipment plus hauling an empty trailer from surplus facility to deficit facility or a combination of two or more hauling empty trailer jobs.

#### 1.0.4 Thesis Outline and Research Contributions

Our focus in Chapter 2 is on the Upper Confidence Bound (UCB) algorithm, which has been known to be effective in balancing the exploration and exploitation trade-off based on a confidence boundary (Lattimore and Szepesvári [2020], Zhu et al. [2019]). The UCB algorithm was proposed by Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer at 2002 (Auer et al. [2002a]). There are different variants of the UCB algorithms, such as UCB1, UCB2, KL-UCB, lil' UCB (Mannor and Tsitsiklis [2004], Karnin et al. [2013], Abbasi-Yadkori et al. [2011], Audibert et al. [2010]). The UCB1 (Auer et al. [2002b]) variant tries each arm once, and at each time step  $t$ , it picks  $A^t = \arg \max_{a_k} \left[ Q_t(a_k) + \sqrt{\frac{\ln t}{N_t(a_k)}} \right]$ , where  $N_t(a_k)$  is the number of times action  $a_k$  has been selected before time  $t$ , and  $Q_t(a_k)$  is the estimated value of  $\theta_{a_k}$ .

The UCB1 algorithm is also known as optimism under uncertainty. It selects the best

arm based on an optimistic estimate and achieves regret that increases only logarithmically with the number of actions taken. This algorithm assumes that given the observations so far, each arm is as good as it can possibly be. The best arm is chosen based on this optimistic estimate.

In Chapter 2, we propose an optimal incentive dynamic pricing mechanism using a UCB-based model to reduce an imbalance problem in a trailer-sharing system subject to a budget limitation using a regret minimization approach.

Our proposed incentive-pricing mechanism and the Convoy’s flexibility encourages drivers to switch their drop-off facility to a location with a larger trailer deficit, which contribute to balancing Convoy’s facilities. In Chapter 2, we also formalize the facility selection algorithm. While we focus on trailer-sharing in the trucking industry, similar problems arise in the other domains, such as car/bike/scooter sharing. In car-sharing (e.g., car2go, ReachNow, Zipcar) and bike-sharing (e.g., CitiBike), the similar capability is offered that customers select where to drop-off vehicles/bikes besides the location where they picked up the vehicle/bikes.

The contributions of Chapter 2 are as follows:

1. Linking pricing and multi-armed bandit problems in the online market to design dynamic pricing under a budget constraint under minimal assumptions about the drivers’ costs distribution.
2. Developing understanding of the balancing problem, particularly regarding the relocation requests by Convoy drivers that are made without any prior planning or reservation, which is a phenomenon unrepresented in the literature.

An alternative to the dynamic pricing mechanism is some type of auction pricing mechanism, in which the algorithm allows platforms (players) to decide on the price and offer it

to the workers (drivers). In an auction mechanism workers submit their bids to a platform that leads workers' preference discovery (willingness to gain for their services) and results in higher revenue. Therefore, the final price in a well-designed auction pricing mechanism can reflect the customers' true preferences. In Chapter 3 we designed an optimal incentive auction pricing mechanism to maximize the cumulative profit via empty trailer return in a trailer-sharing system.

Wang [1993] showed that when customer preferences are sufficiently scattered the auction pricing mechanism tends to be preferred than dynamic pricing mechanism. Literature shows auction pricing mechanisms have been used to find the lowest cost supplier (Mukhopadhyay and Kekre [2002]) and match customers (as sellers) selling their own goods or services to other customers (as buyers) (Simonsohn and Ariely [2008]). One of the advantages of auction pricing mechanisms is that with modern technologies it can be implemented quickly and generate revenue much faster than dynamic pricing. Since both auction and dynamic pricing mechanisms have their own limitations, a hybrid solution, which enables firms to add another customer segment, has recently been used in operations management and marketing areas to increase the revenue of the firm (Etzion et al. [2006], Hann and Terwiesch [2003], Fay [2004], Terwiesch et al. [2005], Spann and Tellis [2006], Amaldoss and Jain [2008], Wang et al. [2009]). Priceline.com was the first to implement a hybrid pricing mechanism (Kasavana and Singh [2001]). Gneezy et al. [2012] use a relatively simple strategy in a hybrid pricing mechanism: the firm assigns a fixed threshold, and accepts and rejects all bids above and below this threshold, respectively. Since the customers do not know this threshold amount, they may bid above the threshold and the firm can earn additional revenue.

Chapter 3 of this thesis is motivated by the above threshold method in auction. The key idea in this chapter is to pretend that instead of budget on resources we have costs

for using them. That is, we assume that each driver (resource) can be used at a specific cost per unit. These costs are not known to the platform. Then, we have the “resource utilization cost” of each driver (in a bandit framework, arm). We want to choose a driver which generates more reward at less cost. The problem is, we do not know the reward of each driver and we really do not know the cost. Instead, we approximate the ratio of reward over cost per driver. This means, we seek a driver which maximizes this ratio. Therefore, we approximate this ratio (known as bang-per-buck ratio) using the principle of optimism under uncertainty. We design an algorithm inspired from reverse second-price auction mechanism using bang-per-buck ratio (UCB-based reward to the corresponding bid) that could be applied to any crowdsourcing platforms. Our proposed auction pricing mechanism takes the submitted bids by drivers as inputs, selects a driver as winner by considering the bang-per-buck ratio selection criteria, and sets the payment (motivated by second-price auction) to the driver. Sometimes drivers do not trust the requester to reveal their true cost, or the cost may not even be known to a driver. Communicating these true costs to the requester may be challenging in real-world settings. There is some research (Wu et al. [2019], Braverman et al. [2019], and Biswas et al. [2015]) that considers rational workers in their model, which means workers strategically put their bid to maximize their payoff (the difference between the received payment and its cost). Therefore, we are interested in designing a truthful auction pricing mechanism in the presence of rational drivers to report their bids equal to their costs to maximize the cumulative profit via empty trailer return. Our proposed auction mechanism also ensures that the winning driver has a non-negative payoff due to individual rationality, because otherwise, the driver will reject the offer.

Similar to our previous UCB-based algorithm, we needed to solve an optimization problem to get the near-optimal arm. We then analyzed the regret of our auction pricing mechanism by simulating and comparing the regret of the two proposed pricing algorithms- dynamic pricing in Chapter 2 and auction in Chapter 3. We prove that our bidding model is empiri-

cally efficient compared to the dynamic pricing mechanism and dramatically outperforms it. We also prove the truthfulness and individual rationality of our auction pricing mechanism. Our experiments support the practical applicability of our mechanisms on any crowdsourcing platforms to increase its efficiency. We start with a mechanism for the dynamic pricing and then extend it to our auction pricing mechanism.

Chapter 3 makes the following contributions:

1. Combining the auction pricing system into the UCB-based model, using bang-per-buck ratio,
2. Solving both winning arm selection and payment computation problems, in the presence of rational drivers (arms),
3. Executing simulation to compare the performance of auction and dynamic pricing mechanisms. We find that auction pricing algorithm outperforms dynamic pricing algorithm.

In Chapter 4, we discuss the application of machine learning and natural language processing (NLP) in healthcare. Multiple myeloma is a cancer of antibody-secreting bone marrow plasma cells, with over 30,000 new cases diagnosed annually in the United States. The disease causes extensive skeletal destruction, resulting in painful fractures, kidney failure, frequent infections, and anemia, causing severe fatigue. The average age at diagnosis is 69 years, and African American people are disproportionately affected when compared to Caucasian people. Although the average life expectancy after diagnosis is 5 years, recent advances in the development of new therapies are continuously improving outcomes. Currently, there are 19 approved drugs, from six different pharmacologic classes, that may be given in 28 recognized combinations. However, selecting which drug to offer a patient that minimizes toxicity while maximizing benefit is challenging. While clinical trials are designed

to inform which drugs may be more efficacious within a population of patients, they are less informative when making individualized decisions.

In collaboration with the Fred Hutchinson Cancer Research Center in Seattle, Washington, we are proposing to address this unmet need by employing state-of-the-art machine learning methods to help make predictions about which clinical features most accurately predict treatment toxicity (infection) and efficacy (time-to-next-treatment). Traditionally, the first step when addressing this problem would be to create a database of clinical features through a time-intensive and tedious process of manual abstraction. In doing so, the investigator must preselect which features to abstract based on prior knowledge of the disease and treatment. However, this process may introduce unwanted bias and potentially miss unexpected features that prove to be more accurate at making the decision. NLP has proven to be crucial in lessening the time investment needed to simply construct a dataset. Ideally, an algorithm would be applicable to any type of medical records and would extract all relevant data needed for the study without need for human intervention. If perfected, this would reduce the data collection phase to a small fraction of what it is today and would provide an automated solution to structure the data. Unfortunately, this does not yet exist, but NLP has been successfully used in tandem with manual reviews to enhance the process. NLP has been applied in a wide range of clinical applications, such as being used to scrape published literature to explore connections between cancer and certain genes, extract relevant data from clinical records, and predict clinical outcomes. This shows promise that, with more development, NLP will be able to contribute more extensively, and eventually take over the data extraction process. Therefore, we are proposing to use Amazon Comprehend Medical to leverage natural language processing to extract all entities from medical records and use those in an unbiased way to build a model to predict treatment efficacy and toxicity. To the best of our knowledge, this paper is the first to assess the reliability of ACM text and sentiment analysis which is a commonly used and available cloud-based tool.

The contributions of Chapter 4 are as follows:

1. Using of automated cloud-based sentiment analysis (Amazon Comprehend Medical Services) for quality assessment of electronic health records for physicians.
2. Disagreements between the pooled sentiment assigned by the human extractors and the sentiment assigned by the automated Amazon Comprehend Medical Services were qualitatively and quantitatively assessed.

Chapter 5 provides a summary of our research works in this thesis and potential future research directions in crowdsourcing and online on-demand platforms, as well as medical data extraction for treatment of cancer and other chronic diseases.

## Chapter 2

### DYNAMIC PRICING MECHANISM FOR TRAILER-SHARING

#### **2.1 Introduction**

In recent years, through the growth of the Internet, numerous sharing economy platforms with various services (such as TaskRabbit, Upwork, Etsy, Fiverr, Click Worker, Airbnb, Uber, Lyft, Getaround, CitiBike, Convoy, etc.) have emerged and offered a tremendous amount of jobs, products, and services. These platforms minimize costs. Such behavior of digital platforms (acting in their interest) might affect their usefulness. For example, in a trailer-sharing system, drivers typically prefer to drop off empty rental trailers at the last destination of their job because they may want to arrive home sooner or improve productivity by getting to the next expensive load (could be an off-network load of Convoy <sup>1</sup>). In contrast, a rental company that wants to maintain more balanced facilities asks drivers to return empty trailers to the original locations. Later, we discuss how we dealt with this challenge. On these online platforms, users can earn money for job completion. Usually, the platform needs to figure out a payment plan for users (workers) to maximize the utility derived from the job completion. Workers are rational and try to maximize their utility by deciding which job to do and at what price.

Overpricing the jobs would quickly exhaust the platform's budget, whereas underpricing would result in more job rejection and worker shortages for job completion. Therefore, one of the critical parts of these platforms is to design a monetary incentives pricing mechanism that will achieve the company's goal, which could be maximizing the profit, minimizing the cost, maximizing the utility, maximizing the service quality, etc. One way to set prices for

---

<sup>1</sup>Off-network loads means any jobs from other platforms.

such platforms is to estimate workers' costs and then devise an optimal fixed price that would maximize the utility of the online platform. However, there are many challenges in deriving this optimal price. For example, strategic workers, may misreport their actual costs to maximize their utility, and the online system <sup>2</sup>. In this chapter, we show how to design a pricing mechanism that utilizes optimal pricing policies by using the approach of regret (the difference between the expected cumulative rewards of an optimal strategy and the employed strategy) minimization in online learning. In this pricing mechanism, we assumed the online platform does not have access to the drivers' costs. The mechanism sequentially interacts with drivers and adjusts the offered price by using the observed acceptance or rejection feedback from the past drivers.

In this chapter, we evaluate our pricing algorithm for Convoy in simulations. Convoy, a Seattle-based start-up which moves millions of truckloads each day through its network of carriers, is like Uber for trucking. It is solving problems in the \$800 billion trucking industry. Truck drivers use Convoy's app to easily find shipment jobs<sup>3</sup>. Convoy Power Only (CPO) enables truck drivers to haul pre-loaded trailer (drop-and-hook freight). Convoy supplies the trailers from its universal trailer pool and drivers just need their cap of the truck (tractor). In booking a shipment job, the driver selects the desired pickup and drop-off cities and the date of picking up the pre-loaded trailer. Convoy then offers a list of shipments and sets the prices for these choices. The driver can now either accept the job at the offered price or bid on the offered price (see Figure 2.1).

After the Convoy Power Only driver chooses a shipment job, they arrive at the pickup facility and hook up to the pre-loaded trailer. During the job, he completes either a live load/live unload or a drop-and-hook, which means he drops off the pre-loaded trailer and

---

<sup>2</sup>There are three major challenges to design an online pricing mechanism: 1. data uncertainty, 2. economic robustness, and 3. revenue maximization with incomplete information.

<sup>3</sup>Convoy amassed over 500 shippers on its platform. The leading shippers include The Home Depot, Niagara, P&G, Land O Lakes, Unilever, Dollar Shave Club, Ardagh Group, Anheuser-Busch, CHEP, GE Appliances, Treasury Wine Estates, Dollar General, LG Electronics, etc.

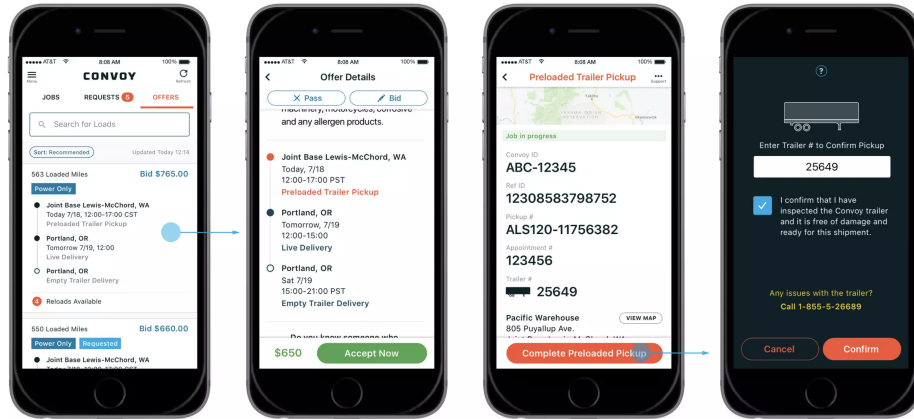


Figure 2.1: Booking a shipment job [Convoy-Team, 2020]

hooks to another pre-loaded trailer). Then, he must return the empty trailer to the origin location/city, which is called an empty trailer return (ETR) (see Figure 2.2).

Each Power Only load that drivers bid on has a pre-assigned empty trailer return location (often back to the origin facility or another facility in the same city). Though this helps ensure more equal distribution of trailers, it limits drivers' ability to get to their next load. As we mentioned above, our business partner, Convoy, wanted to satisfy<sup>4</sup> its drivers by increasing the drivers' ETR flexibility to be a more useful platform. Hence, it announced "Go Plus", a key enhancement to its nationwide drop-and-hook marketplace by offering flexibility to drivers on empty trailer returns for dropping off trailers at their desired locations. Therefore, at his last facility, the driver can send a request to Convoy to drop off the empty trailer at his desired facility or somewhere closer to it than the initial assigned drop off location, with a maximum bob-tailing (driving a truck without a trailer attached) distance constraint. This request is called a "relocation request" (see Figure 2.3). Bob-tailing is detrimental to the driver because it accelerates the depreciation rate of a truck, and increases the risk of locking up, skidding and losing traction due to the lack of the

<sup>4</sup>Convoy conducted a carrier survey in March 2021 and had the following takeaways. First, it confirmed the high carrier desirability about trailer return flexibility. 57% of respondents indicate a desire for both time and location flexibility. Second, "Getting to the next load" is the primary reason why drivers like flexibility.

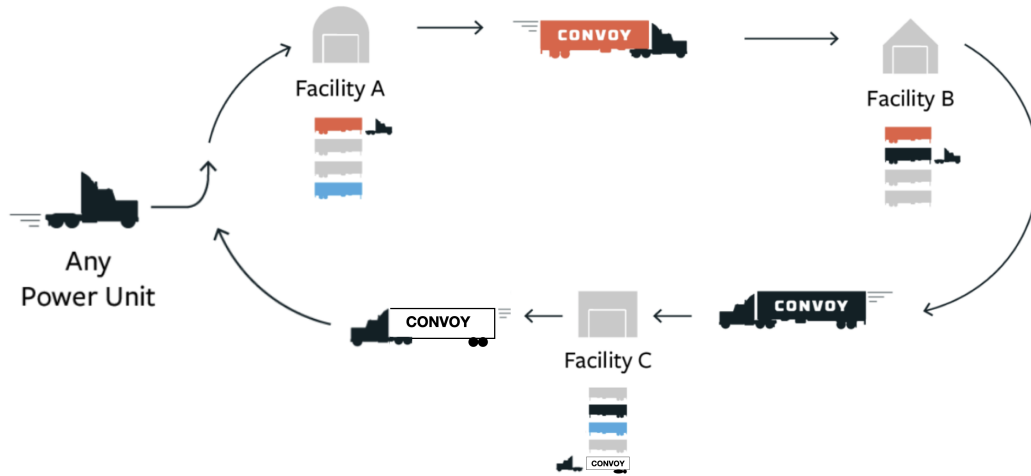


Figure 2.2: A shipment job [Hubert, 2019]

weight, specifically in wet road conditions. Therefore, drivers set a maximum bob-tailing distance for their ETR, which differs from driver to driver.

With the launch of Go Plus, each Power Only load has not only the default empty trailer return option, but also a list of alternatives. After receiving the relocation request by considering the maximum bob-tailing constraint, Convoy is responsible for choosing a set of facilities with corresponding prices to offer to the driver (see Figure 2.4). Drivers can review the locations to pick the one that works for them most.<sup>5</sup>

The added flexibility of dropping off an empty trailer to another facility can, however, create imbalance problems in Convoy’s system, such as the unavailability of trailers for the next shipments and full parking slots at facilities. The number of requests is dynamic and unpredictable. An imbalance problem is a major challenge in on-demand and crowd economy platforms, yet it is not well studied.

---

<sup>5</sup>Convoy incubated this idea in the first quarter of 2020 on two lanes in the Southwest (SW) and Midwest (MW) regions. The primary takeaway was that it received a lot of interests from drivers about this feature (e.g., 64% of the requests are interested in changing empty trailer return), especially in the SW region where there are many off-network loads.

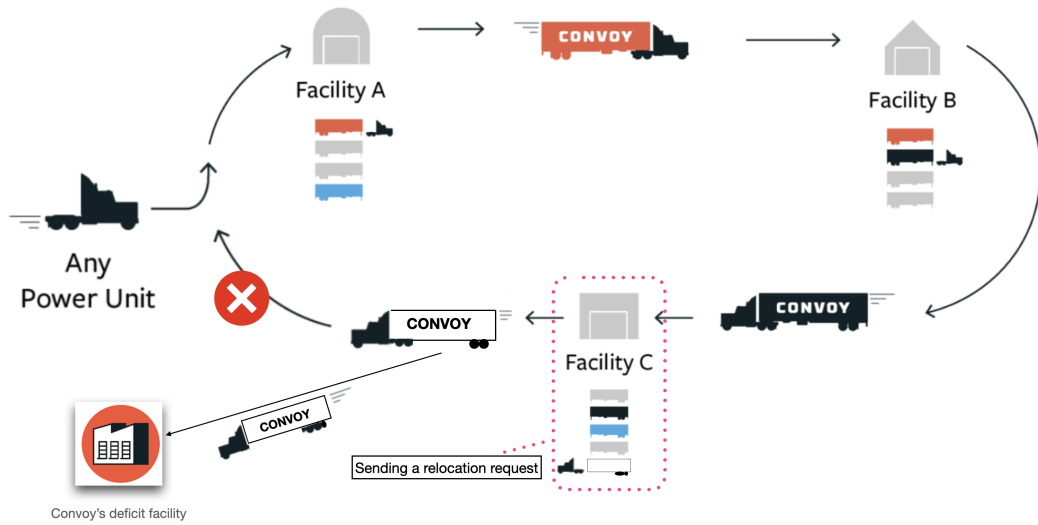


Figure 2.3: GoPlus: ETR flexibility

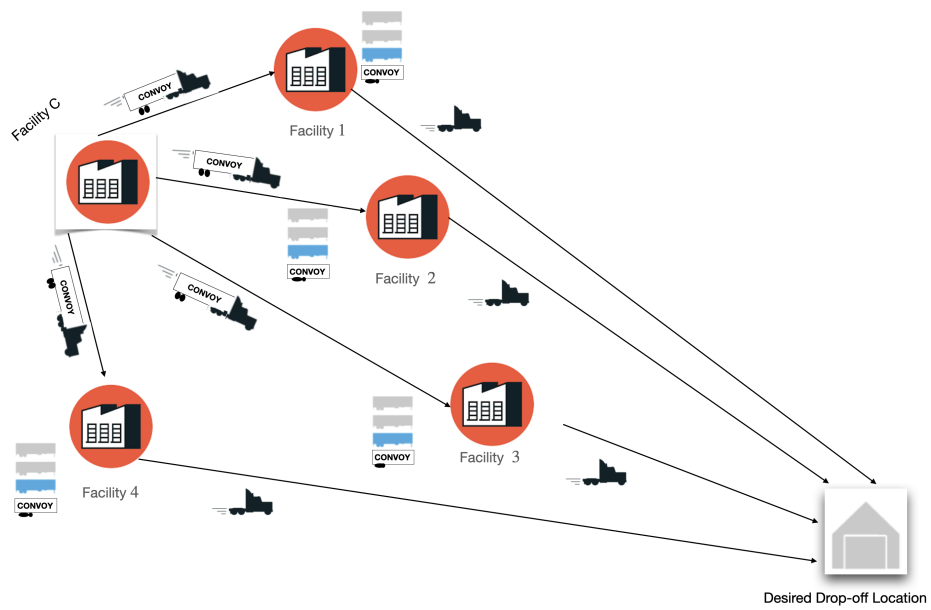


Figure 2.4: Relation between driver's desired location and relocation request

In this chapter, motivated by real crowdsourcing jobs in the digital freight industry, we address the following questions for Convoy:

1. What price should Convoy offer a driver for rebalancing facilities in an online market?
2. How should Convoy engage drivers in making decisions about their drop off locations to reduce the imbalance problem?

We attempt to contribute to the developing understanding of the balancing problem, particularly regarding the relocation requests by Convoy drivers that are made without any prior planning or reservation, which is a phenomenon unrepresented in the literature. To solve this problem, we built a facility selection algorithm that interacts with the drivers, decides which facilities to offer and their incentive prices, and learns from past interactions. Our dynamic pricing model aims to maximize Convoy’s utility, defined as the number of offer acceptances. In setting prices, there is an apparent trade-off between offering the highest price (although this price maximizes the number of job acceptance, it quickly exhausts the budget) and offering the lowest price (it is less effective in terms of offer acceptance and results in minimizing the utility).

Multi-armed bandits is a robust framework for algorithms that make decisions over time under uncertainty. In dynamic pricing problems (a.k.a optimizing seller’s prices or learn-and-earn), when a new customer arrives, the seller chooses a price offered to the customer. The customer purchases or does not purchase. The seller’s goal is to maximize the total profit. Therefore, dynamic pricing problems fall under the multi-armed bandit framework. Our main contribution to the study of monetary incentives (any financial incentives to motivate workers towards completing their task) is linking pricing and multi-armed bandit problems in the online market to design dynamic pricing under a budget constraint<sup>6</sup> under

---

<sup>6</sup>In the maximum task coverage problem, the first and important variant is when a budget constraint is given for each time period or for the entire campaign (To et al. [2016]).

minimal assumptions about the drivers' costs distribution.

This research contributes to the literature of dynamic pricing by 1. formalizing a problem that captures the pricing model faced by many online labor markets or even online markets in general; and 2. proposing a novel pricing mechanism that uses the responses of drivers effectively and achieves low regret with respect to time. Our pricing mechanism is dynamic (online learning characteristic) and can decide what incentive prices to offer to each selected facility.

## **2.2 Literature Review**

In this section, we review the related works to our research. There are different lines of investigation that have various overlap with our work.

### *2.2.1 Truckload Brokerage Services*

With the growth of e-commerce applications for transportation, there is an extensive literature on the use of web-based or app systems for developing brokerage services. [Luncean et al., 2014] showed that providing automated support of allocation freight requests to carriers increases the degree of automation of business activities. [Scheuermann and Hoxha, 2012] extended this stream of research by combining semantic web technologies and service-oriented computing. The existing literature mainly focuses on facilitating the connection of the shippers with the carriers. [Tsai et al., 2011] provided a closed-form pricing model for basic truckload options (uncertainty in transportation capacity and cost using derivative contracts) that could be valuable to both the shippers and the carriers. [Lindsey et al., 2013] studied online pricing model and capacity sourcing for third-party logistics providers. They used optimization to determine the optimal price for shippers that maximizes the third-party's expected profit. In addition, their models determined a list of potential carri-

ers for an incoming load and also a ranked carrier list.

To the best of our knowledge, none of the existing work in transportation explicitly addresses the automated mechanism for online pricing and dynamic facility selection between brokerage and carriers. In our work, we studied dynamic pricing from the perspective of freight brokerage, which proposes a fixed price and then observes the response of the carriers.

### *2.2.2 Incentivizing Workers in Crowdsourcing*

There is an extensive literature in understanding the right incentives for workers in the online market, including monetary and non-monetary incentives ([Tokarchuk et al., 2012]). [Horton and Chilton, 2010] and [Auer et al., 2002a] studied workers' proper wages and payment rates, respectively. Also, [Mason and Watts, 2009] and [Shaw et al., 2011] studied the quality of service that improves the workers' performance. In the literature, several challenges were considered, such as the platform's large scale, request (demand) fluctuation over time, unknown individual costs of drivers and their strategic behavior, and the budget constraint. [Qiu et al., 2019] proposed a pricing policy and a rating policy to define how a requester should determine a fair payment for crowdsourcing workers according to worker performance. [Faullant et al., 2013] found that workers are more likely to engage with crowdsourcing jobs if they feel that they are being treated fairly, and their payment is corresponding with their contributions.

In this chapter, we explore how we can engage drivers to solve the imbalance problem in trailer sharing in exchange for monetary incentives. We design an incentives pricing mechanism in a crowdsourcing platform that incentivizes the drivers in the process of repositioning of trailers by providing them with alternate choices to return trailers to in exchange for monetary incentives.

### 2.2.3 *Dynamic Pricing Model*

Recent papers have addressed pricing in the context of labor platforms. Regarding comparing different pricing schemes [Cachon et al., 2017] showed that dynamic pricing is better for all stakeholders. [Banerjee et al., 2015] studied a stochastic queueing model with price sensitive riders and drivers and showed that static pricing is asymptotically optimal. [Bimpikis et al., 2019] examined the platform decisions with respect to the ride pricing and driver compensation across a network. They assumed that drivers behave strategically and performed an equilibrium analysis.

As opposed to peer-to-peer product sharing (e.g., [Benjaafar et al., 2019]), our platform is an on-demand platform (that deals with non-differentiated jobs i.e., any worker can do any job). [Horton and Chilton, 2010] proposed a model of workers in order to estimate their wages, and [Horton and Zeckhauser, 2010] investigated an automated way to negotiate payments with workers. In our setting, in contrast, the drivers' costs are unknown, and the budget is employed only in time steps when the offer is accepted by the driver<sup>7</sup>, which none of the standard MAB approaches apply to this setting. Our dynamic pricing mechanism uses the regret minimization framework.

### 2.2.4 *Dynamic Fleet Repositioning and Balancing Problem*

The problem of imbalance and repositioning of the fleet has been extensively studied in vehicle sharing systems, including cars, bikes, and scooters ([Kek et al., 2009]; [Nair and Miller-Hooks, 2011]; [Cepolina and Farina, 2012]; [de Almeida Correia and Antunes, 2012]; and [Clemente et al., 2013]). [George and Xia, 2011], [Schuijbroek et al., 2017], [Pfrommer et al., 2014] studied bike repositioning by deploying fleets of trucks. [Fricker et al., 2012], [Chemla et al., 2013], and [Pfrommer et al., 2014] showed an improvement in the balancing

---

<sup>7</sup>Since, otherwise the driver will return the empty trailer to the original assigned facility in the accepted job.

of the system by offering incentives to the users for contributing in balancing the sharing system.

[Chemla et al., 2013] and [Raviv and Kolka, 2013] proposed repositioning policies under a fixed daily budget constraint. [Goel and Mehta, 2008], [Aggarwal et al., 2011], [Devanur and Hayes, 2009a], and [Devanur et al., 2013] studied the theory of online algorithms for matching and allocation problems in the crowdsourcing setting.

Our work is motivated by a lengthy series of papers on the dynamic vehicle allocation problem and managing fleets of vehicles (Alonso-Mora et al. [2017], Turan et al. [2020], Ströhle et al. [2019], Simonetto et al. [2019]), but we assume that we make decisions to move trailers at time  $t$ , and then learn information about new drivers at time  $t + 1$ . All the prior models addressed the problem of designing an effective load acceptance mechanism for making these assignments, before the loads can be actually dispatched in the future. Our facility selection algorithm is dynamic and considers the current state of the system to decide which facilities to offer.

### *2.2.5 Multi-armed Bandit Problems*

[Auer et al., 2002a] introduced one of the MAB algorithms which is the upper confidence bound (UCB1) algorithm, and [Kleinberg, 2004a], [Auer et al., 2007], [Besbes and Zeevi, 2009], and [Srinivas et al., 2010] extended this approach to improve the regret bound. Other MAB algorithms were proposed by ([Auer et al., 2002a]; [Bubeck and Cesa-Bianchi, 2012]; [Filippi et al., 2010]; and [Chapelle and Li, 2011]) to model sequential decision making in an uncertain environment. Badanidiyuru et al. [2012] studied crowdsourcing settings by using the MAB algorithm. They presented a fixed-price mechanism that was designed for stochastic setting in which the players are identically distributed. They combined online learning with secretary problems. We omit a detailed review of this area in the interest of

space.

To et al. [2016] showed that in the maximum task coverage problem, the first and important variant is when a budget constraint is given for each time period or for the entire campaign. In budgeted MAB, besides the random reward, each arm is also associated with an unknown random cost. During each round, the player pulls an arm, receives a random reward, and pays a random cost, until his budget runs out. Recently, the budgeted MAB problem has been utilized to model some industrial situations under different settings, for example, online bidding optimization ([Amin et al., 2012a] and [Tran-Thanh et al., 2014]) and in cloud computing ([Agmon Ben-Yehuda et al., 2014]; and [Ardagna et al., 2011]). [Tran-Thanh et al., 2010] proposed the budget-limited MAB  $\epsilon$ -first algorithm, in which the first  $\epsilon$  of the budget is used for exploration to learn the arms' rewards, while the remaining  $1 - \epsilon$  is used to maximize the reward based on those estimates. [Tran-Thanh et al., 2012a] introduced two new algorithms, KUBE and fractional KUBE, for the budgeted MAB problem. Both algorithms sample each arm in an initial phase, and at each subsequent round, they determine a best set of arms. [Vanchinathan et al., 2015] designed the GP-SELECT algorithm to select high-value subsets from a large pool of items. [Ding et al., 2013] studied the MAB problems with budget constraint and variable costs. They proposed two UCB-based algorithms: their first algorithm needs prior knowledge about the lower bound of the expected costs and the second algorithm relaxes this requirement by estimating the minimal expected costs from empirical observations. [Xia et al., 2016] studied the multi-play budgeted linear/contextual bandit in which pulling an arm receives both a random reward and a random cost. Also, [Xia et al., 2015] proposed the best arm identification problem for budgeted MAB in a stochastic environment.

In this study, we link dynamic pricing and multi-armed bandits (UCB1 algorithm) to design a mechanism that is budget feasible and makes minimal assumptions about the distribution

of drivers' costs. As we mentioned, in our MAB setting, in contrast, the costs are unknown, and the budget is limited.

### *2.2.6 Regret Minimization*

The main objective in the MAB problems is to design a no-regret algorithm, which means the average regret asymptotically approaches zero over time. Regret minimization has been studied extensively, and [Blum and Monsour, 2007] provides a good overview. While regret minimization was studied by several authors ([Bonald and Proutiere, 2013]; [David and Shimkin, 2014]), [Carpentier and Valko, 2015] is the first to study the pure-exploration problem in the fixed-budget setting.

As we mentioned, our dynamic pricing mechanism uses the regret minimization framework. We established an upper bound on the expected regret. Our analysis also provides essential insights on the role of cost uncertainty to the buyer in the best achievable regret. This will be explained more fully in Section 2.4.

Let us emphasize that our setting deviates from the settings studied in these papers in four main ways: 1. different facilities, 2. budget limitation constraint, 3. drivers' heterogeneity, and 4. unknown drivers' cost distribution.

## **2.3 Problem Setting and Model**

### *2.3.1 Problem Setting for Dynamic Pricing Mechanism*

In the classical MAB problem setting, there are  $K$  independent arms associated with unknown reward distributions. An MAB algorithm operates in discrete time steps. In each time step, the algorithm selects an arm to get a stochastic reward associated with that arm. By experimenting with potentially suboptimal arms, the algorithm explores in order to learn about the optimal arm. Meanwhile, by pulling the arm that appears best, it exploits its

learning to maximize the reward. The goal of the algorithm is converging to the optimal arm quickly so as to minimize the regret. Usual MAB problems consider a finite number of arms from which the player may select at each time step (Lai et al. [1985]). The number of arms is typically much smaller than the number of experiments allowed. Therefore, exploration of all possible options is usually performed and combined with exploitation of the seemingly best arms. Otherwise, the exploration of all the arms would be impossible to achieve.

The main challenge in deciding the payments in our problem is the unknown distribution of the workers' cost. The pricing mechanism interacts with workers sequentially in discrete time steps  $t$ , offering a price  $p^t$  at each time step to worker  $w^t$  and adjusting the estimates of the distribution of the workers' cost based on observed feedback. In order to model our problem in the MAB framework, we discretize the prices by creating a set of  $K$  prices (arms) similar to (Babaioff et al. [2012], Blum et al. [2004]). For these  $K$  prices,  $p_i, i = \{1, 2, \dots, K\}$ , we maintain  $F^t(p_i)$  as an estimate of the cumulative distribution function of workers' costs for price  $p_i$  at time  $t$  ( $p_i^t$ ). At each time step, our mechanisms will pick the price  $p_i^t$  based on some optimization criterion.

In this chapter, we design an optimal incentive dynamic pricing mechanism to reduce an imbalance problem in trailer-sharing system subject to a budget limitation. Now, we describe the setting of our dynamic pricing mechanism. Consider a driver  $w \in W$  at a given facility who sends a relocation request to Convoy for dropping off an empty trailer at  $l^w$  or somewhere closer to it than the initial assigned drop off location. Convoy, after receiving the relocation request from a driver, offers some facilities. In this section, we assume that each task completion (completing the job or shipment by doing an ETR) has unit value. This value equals to the fixed cost penalty that Convoy should pay to a shipper for not doing the shipment on time due to the trailer shortage. Therefore, Convoy wishes to maximize

the number of assignments.

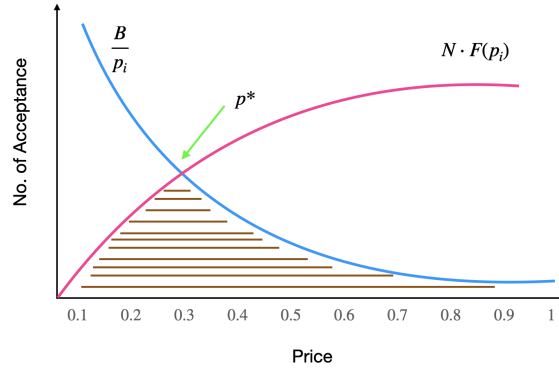
Each driver has a cost  $c^w \in \mathbb{R}$  for completing an empty trailer return (ETR), unknown to Convoy. We assume that costs are bounded,  $c^w \in [\underline{c}, \bar{c}]$ , where  $\underline{c}$  and  $\bar{c}$  are parameters of the pricing mechanism. Drivers' costs are *i.i.d.* sampled, and we define  $F : [\underline{c}, \bar{c}] \rightarrow [0, 1]$  as the cumulative distribution function (CDF) of these private costs. Furthermore, we assume a finite pool of drivers,  $N$  ( $c^w \in \{c^1, \dots, c^N\}$ ).

Now, let us consider an unrealistic full information setting (our benchmark model) in which a player (brokerage) has access to the CDF of drivers' costs. We assume Convoy has a fixed budget  $B$ . Therefore, Convoy wishes to maximize the number of assignments subject to the budget constraint. Let  $\frac{B}{p}$  represent the maximum number of drivers that we can offer at price  $p$  under budget constraint  $B$ .  $N \cdot F(p)$  represents the expected number of drivers that we can get at price  $p$  with unlimited budget. Then, the optimal price  $p^*$  lies at the intersection of these two functions (see Figure 2.5). We denote this optimal benchmark price by  $p^*$ , which is given by

$$p^* = \arg \max_p \min \left\{ \underbrace{N \cdot F(p)}_{\substack{\text{Expected number of} \\ \text{drivers that we can get} \\ \text{at price } p}}, \underbrace{\frac{B}{p}}_{\substack{\text{Maximum number of drivers} \\ \text{that we can offer by the lim-} \\ \text{ited budget } B \text{ at price } p}} \right\} \text{ s.t. } p \in [\underline{c}, \bar{c}]$$

The number of drivers we can get at price  $p$

where  $N$  denotes the number of drivers and  $B$  is the Convoy's budget. Since the pricing mechanism does not have access to the optimal price  $p^*$ , we need to define an alternate optimal price  $p^{\text{Opt}}$  as the best price among the  $K$  discretized prices. Let  $i^{\text{Opt}} \in \{1, 2, \dots, K\}$  denote the index of the optimal price among the  $K$  prices.  $p^{\text{Opt}}$  denotes the corresponding

Figure 2.5: Optimal offline price  $p^*$ 

price to arm  $i^{\text{Opt}}$  and  $F^{\text{Opt}} = F(p^{\text{Opt}})$ .

$$i^{\text{Opt}} = \arg \max_{i \in \{1, \dots, K\}} \min \left\{ N \cdot F(p_i), \frac{B}{p_i} \right\}$$

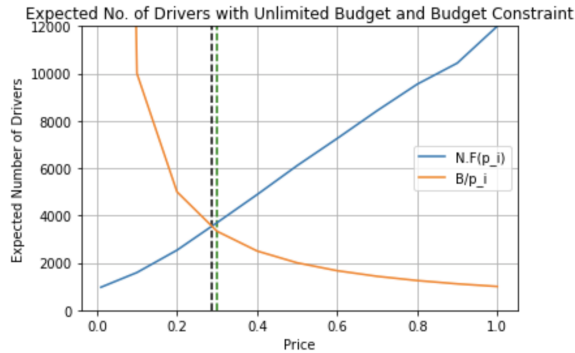


Figure 2.6: The optimal price  $p^*$  lies at the intersection of the  $N \cdot F(p)$  and  $\frac{B}{p}$  curves (black dashed line)<sup>8</sup>. Our mechanism uses discretized  $K$  ( $K = 11$ ) prices<sup>9</sup> ( $p \in \{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ ), and  $p^{\text{Opt}}$  corresponds to the optimal price among these eleven prices (green dashed line).

<sup>8</sup>We used  $p_1 = 0.01$ ,  $P_K = 1$ ,  $K = 11$ ,  $B = 1000$ , and  $N = 12000$

<sup>9</sup>The classical stochastic MAB problems can store only a fixed number of arms in the memory. But some MAB algorithms such as UCB1 (which relies on an initial phase of algorithm) requires all arms to be present in the arm-memory at the start of the algorithm.

We compare our mechanism against our benchmark in an (unrealistic) offline mechanism with complete access to the workers’ true costs. In this setting, we sort the workers by their increasing costs and offering each worker their true cost. The primary objective of this study was to get cost distributions from a realistic job on Convoy platform as well as to understand drivers’ feedback regarding the offered price for the job.

In our empirical result,  $p^* = 0.29018$  and  $p^{\text{Opt}} = 0.30000$ . Since our pricing mechanism does not have access to the optimal price  $p^*$ ,  $p^{\text{Opt}}$  is the best price available.

### 2.3.2 Dynamic Pricing Mechanism

The main challenges in deciding the payments in our problem are the unknown distribution of the drivers’ costs ( $F$ ), as well as pool size of drivers ( $N$ ), and the limited budget ( $B$ ). Similar to the classical  $K$ -armed bandit problems, we create a discrete set of  $K$  prices,  $\{p_1(= \underline{c}), \dots, p_K(= \bar{c})\}$ , to formalize our problem in the MAB framework. For these  $K$  arms, we maintain  $F^t(p_i)$  as an estimate of the CDF of drivers’ costs for price  $p_i$  at time  $t$ . At each time step  $t$ , the pricing mechanism pulls arm  $i^t$  and makes a price offer  $p_i^t$  to driver  $w$ . The driver accepts or rejects the offer. The pricing mechanism updates the CDF of the driver’s cost based on received feedback. The pricing mechanism stops execution when the budget is exhausted.

While the typical goal for MAB algorithms is to maximize cumulative reward or minimize regret (a.k.a pure-exploration bandits), an alternative goal is to output the best arm (choice) after each round  $t$  (a.k.a best-arm identification). One of the good algorithms for this version of bandit problems is UCB1 (Bubeck et al. [2009]).

Our solution is inspired by the classical UCB1 algorithm, where we consider exploration (learning about different, potentially suboptimal, prices) and exploitation (offering the price

that appears the best price so far).

To describe our pricing mechanism, we need to define some additional notations. Let  $B^t$  be the remaining budget at time step  $t$ , and  $n_i^t$  be the number of times price  $p_i$  has been offered by time  $t$ . To tackle the limited feedback, we maintain an upper confidence bound on  $F^t(p_i)$ , denoted by  $\tilde{F}^t(p_i) = F^t(p_i) + \sqrt{\frac{2 \cdot \ln(t)}{n_i^t}}$ , as in the UCB1 algorithm. Then, the mechanism picks the best arm:

$$i^t = \arg \max_{i \in \{1, \dots, K\}} \min \left\{ N \cdot \tilde{F}^t(p_i), \frac{B}{p_i} \right\}.$$

$p_{i^t} = p^t$  denotes the corresponding price for arm  $i^t$ , which the mechanism offers as a “take it or leave it price” to the driver. The pricing mechanism receives the binary feedback  $y^t$  from the driver:

$$y^t = \begin{cases} 1, & \text{the driver accepts the offered price } p^t \text{ (when } c^w \leq p^t) \\ 0, & \text{otherwise,} \end{cases}$$

This is then used to update the parameters only for the arm  $i^t$ . The pricing mechanism updates the following variables:

$$\begin{aligned} B^{t+1} &= B^t - (p^t \cdot y^t) \\ n_{i^t}^{t+1} &= n_{i^t}^t + 1 \\ F^{t+1}(p_i) &= F^t(p_i) + \frac{(y^t - F^t(p_i))}{(n_{i^t}^t + 1)} \end{aligned}$$

The final update formula comes from a recursive version of updating  $Q$  from the Bernoulli MAB algorithm (see Sutton and Barto [2011]). After all the above updates, the pricing mechanism waits for a new driver to place a relocation request.

In this chapter, we are following [Sutton and Barto, 2011, page 31] notations. In a  $K$ -armed bandit problem, let  $A_t \in \{a_1, a_2, \dots, a_K\}$  be an action (arm) selected at time step  $t$  and  $R_t$  be a reward at time step  $t$ :

$$R_t = \begin{cases} 1, & \text{with probability } \theta_k \text{ if } A_t = a_k, k \in \{1, 2, \dots, K\} \\ 0, & \text{otherwise,} \end{cases}$$

The value of action  $a_k$  is defined as follows:

$$\begin{aligned} q^*(A_t = a_k) &:= \mathbb{E}[R_t | A_t = a_k], \quad k \in \{1, 2, \dots, K\} \\ &= \theta_k \end{aligned}$$

where  $\theta_k$  is unknown to the player. Let  $Q_t(A_t = a_k)$  be an estimated value of  $q^*(A_t = a_k)$ , which is given by:

$$\begin{aligned} Q_t(A_t = a_k) &:= \frac{\text{sum of rewards when } A_t = a_k \text{ taken prior to } t}{\text{number of times } A_t = a_k \text{ taken prior to } t} \\ &= \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a_k}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a_k}}, \quad \forall k = 1, 2, \dots, K \end{aligned}$$

### 2.3.3 Problem Setting for Facility Selection Algorithm

As we mentioned above, our incentive pricing mechanism encourages drivers to switch their drop off facility to a facility with a larger trailer deficit to contribute to balancing Convoy's set of facilities. We now explain how our dynamic facility selection algorithm determines a set of offered facilities for ETRs.

The population of these drivers, which induces the cost distribution  $F$ , are drivers who are at a facility right now, waiting for a live unload or a drop (a pre-loaded trailer)-and-hook (an empty trailer), and who are willing to do a relocation. Consider a driver  $w$  who is at a given facility and is interested in location  $l^w$ . He sends a request to Convoy for

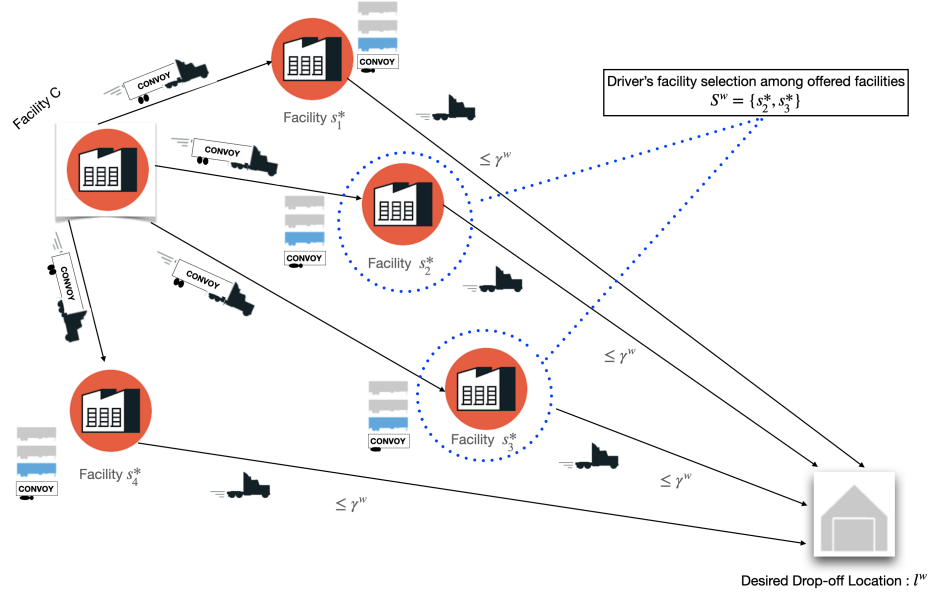


Figure 2.7: Facility Selection

dropping off the empty trailer at  $l^w$  or somewhere closer to it than the initial assigned drop off location (relocation request). We assume that the driver  $w$  is willing to bob-tail (driving without trailer) up to a maximum distance  $\gamma^w$  from  $l^w$ . For simplicity, we assume for any distance up to  $\gamma^w$  the driver's cost  $c^w$  is constant (see Figure 2.7).

#### 2.3.4 Facility Selection Algorithm

In this section, we formalize the facility selection algorithm. Convoy has  $m$  facilities  $S = \{s_1, \dots, s_m\}$ . When driver  $w$  sends a relocation request at time  $t$ , our dynamic facility selection algorithm is responsible for determining a set of alternate drop off locations  $S^* = \{s_1^*, \dots, s_z^*\} \subset S$ , where  $z \ll m$ , based on (1.) the reported desired location of the driver ( $l^w$ ), and (2.) the maximum bob-tailing distance of driver ( $\gamma^w$ ). The facility selection algorithm generates option set  $S^* = \{s : d(s, l^w) \leq \gamma^w\}$ . Then, the driver selects 0, 1, or more of these options based on his cost and reports it to Convoy. We denote this collection set by  $S^w$ , where  $S^w \subseteq S^*$  (see Figure 2.7). The pricing mechanism executes on each

facility  $s \in S^w$  and computes the value of the incentive  $p_i^t(s)$ . Finally, we display the offers (including a facility selection option and the corresponding price for each) in descending order in terms of trailer needs.

Convoy can even offer a pair of facilities, including one deficit and one surplus, considering that the summation of all bob-tailing distances should not exceed  $\gamma^w$ . In this way, a driver contributes more to balancing the facilities by dropping off the trailer in a deficit facility, then bob-tailing to a surplus facility, picking up an empty trailer, and dropping it off in one of the deficit facilities in (or close to) the driver's desired destination. Convoy displays all potential offered facilities such that all pair options are shown first in descending order of trailer need; then all single options are shown in the descending order of trailer need. Note that if the number of needed trailers in two or more facilities is the same, we consider the required bob-tailing distance and sort them in ascending order. In this case, Convoy can decrease its costs by offering a lower price with lower bob-tailing distance.

## 2.4 Results

Our dynamic pricing mechanism aims to minimize regret by quickly converging the offered prices to the optimal price. In this section, we conduct through simulations to investigate the performance of the proposed algorithm on simulated cost distributions. We consider uniform simulated distribution in  $[0.1, 0.9]$  for analyzing our algorithm. There are two key parameters: the total number of drivers, and the budget. We will explore the effect of these parameters respectively.

We initialize  $B = 1000$ ,  $\underline{c} = 0.001$ ,  $\bar{c} = 1$ ,  $K = 11$ ,  $N = 12000$ , and  $p = \{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ . The optimal benchmark price is  $p^* = 0.292285577383811$ , and  $p^{opt} = 0.30000000000000004$  (Figure 2.6). We calculate the percent of times the optimal price is chosen at each iteration (100 simulation). Then, we plot the mean rate of choosing the optimal price at each iteration. Also, we calculate the average price chosen through

our UCB algorithm at each iteration (running 500 simulations on it). Finally, we plot both average price and regret chosen at each iteration.

Figure 2.8 shows the percent of times that the optimal price is chosen for each iteration (after running the algorithm for 500 times) in the algorithm. In Figure 2.9, we plot the average price chosen at each iteration and see how quickly our model can converge to an optimal price. Figure 2.10 shows the average regret of the pricing mechanism which represents the performance of the model. It shows that the average regret of our proposed mechanism approaches zero asymptotically over time.

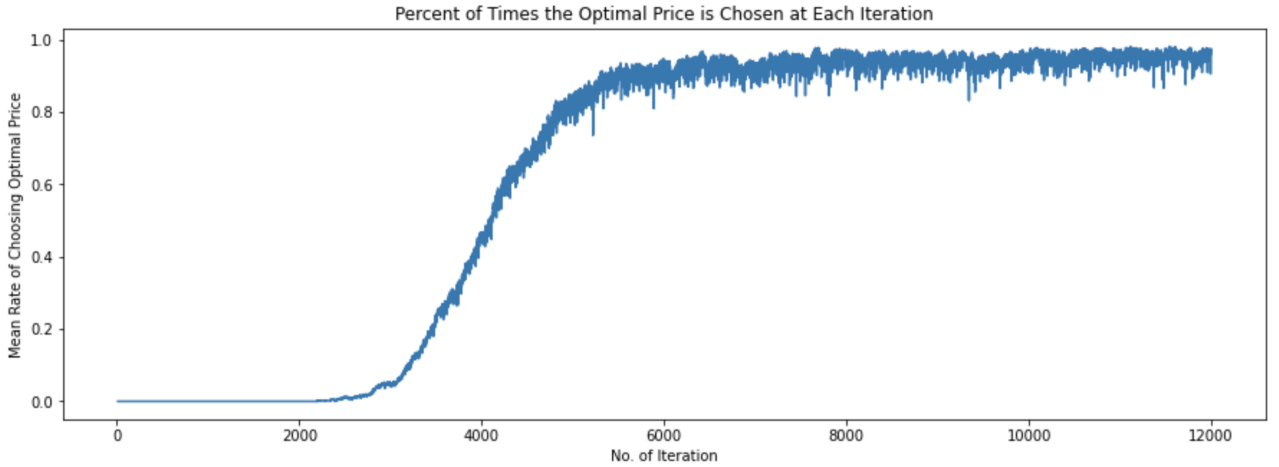


Figure 2.8: Mean Rate of Choosing Optimal Price in 500 Simulation

Figure 2.11 illustrates that our dynamic pricing mechanism performs nearly as well as the full information mechanism, even though it operates under limited information. The optimal utility in the full information setting equals to  $U_{p^*} = \frac{B}{p^*}$ . Also, the expected utility of the pricing mechanism  $M$  is given by  $U_M = \sum_{i=1}^K \mathbb{E}[n_i^T] \cdot F_i$ , where  $n_i^T$  determines the number of times price  $p_i$  has been offered by time  $T$ .

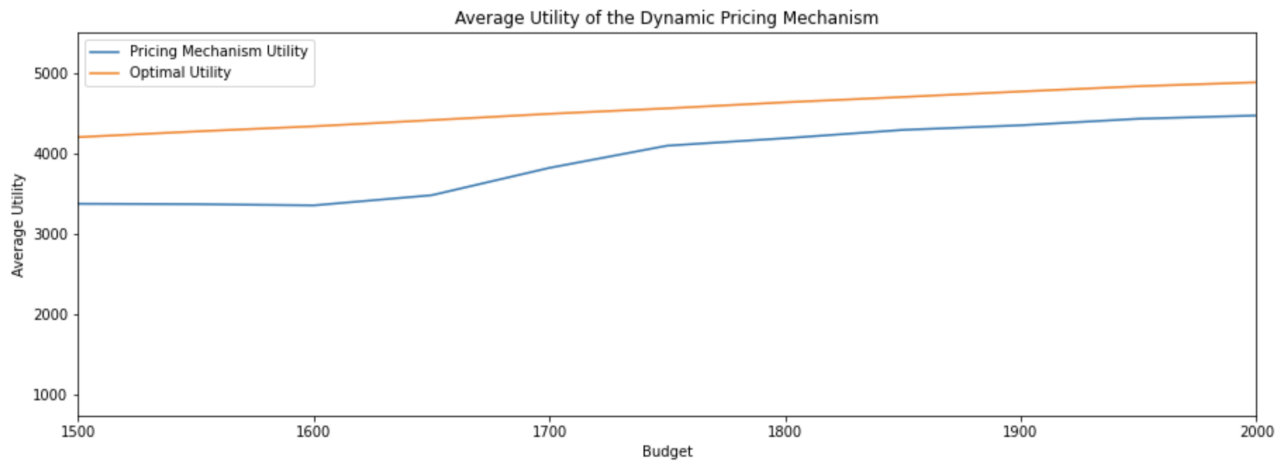


Figure 2.11: Utility for uniform cost distribution in  $[0.1, 0.9]$ , varying budget.

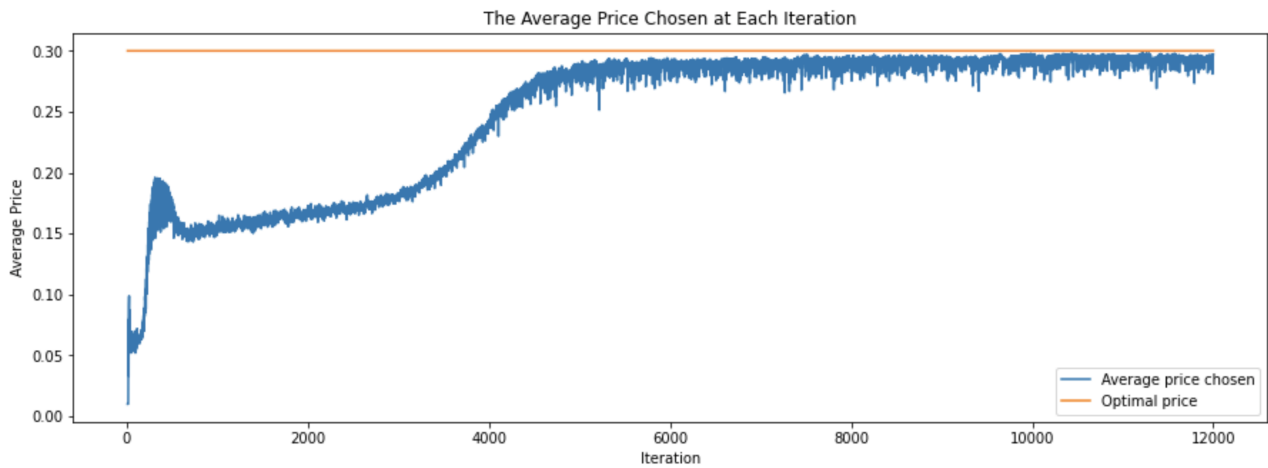


Figure 2.9: Price Convergence

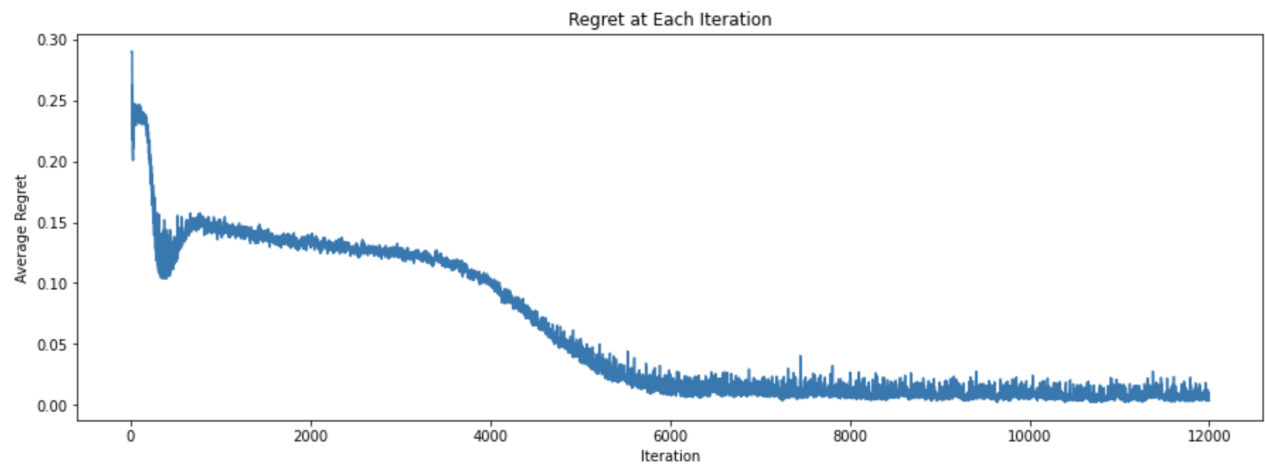


Figure 2.10: Average Regret Convergence

Figure 2.12 shows that the expected average of regret of the pricing mechanism first increases and then decreases in the budget  $B$ . In fact, in our setting underpayment due to a budget constraint, we will have more rejected offers which incurs larger regret. This plot also implies that our proposed dynamic pricing UCB-based algorithm is the method that achieves logarithmic regret bound.

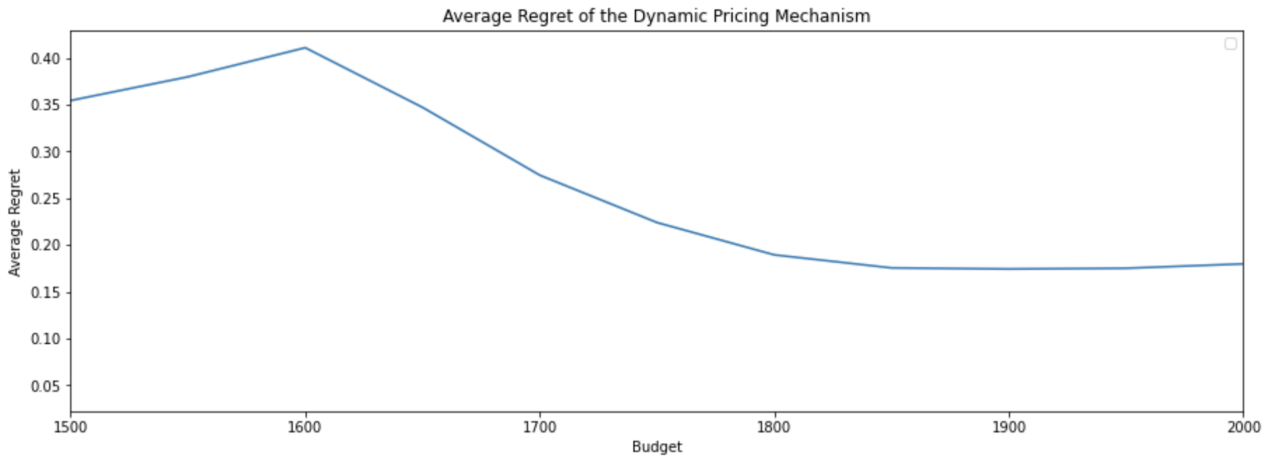


Figure 2.12: Average regret of the pricing mechanisms with increasing budget.

## 2.5 Conclusions and Managerial Insights

In this chapter, we designed a dynamic pricing mechanism to reduce an imbalance problem in a trailer-sharing system subject to a budget limitation using a regret minimization approach. Our proposed incentive-pricing mechanism engages drivers in the facility balancing processes. We collaborated with the digital freight company Convoy and built pricing and facility-selection algorithms for their platform, which can be used in other online markets. We show (by simulation) that our pricing mechanism is empirically efficient compared to the optimal offline market (benchmark) price.

While we focus on trailer-sharing in the trucking industry, similar problems arise in the other domains, such as car/bike/scooter sharing. In car-sharing (e.g., car2go, ReachNow,

Zipcar) and bike-sharing (e.g., CitiBike), companies offer a similar capability that allows customers to select where to drop-off vehicles/bikes besides the location where they picked up the vehicle/bikes.

## **2.6 Future Work**

There are some extensions for future work. Here, we considered a simple utility function for Convoy (buyer). It would be useful to extend our approach to more complex utility functions. Additionally, we consider a maximum bob-tailing factor in facility selection from the drivers' perspectives. One can consider more factors such as the chance of availability of the next shipment in a specific facility, road and weather conditions, etc., in the model. We could also implement this mechanism within real-world crowdsourcing platforms. Based on the studies of the car-sharing industry, the one-way service (i.e., do not need to return the car back to the origin) is more popular for spontaneous short trips. In contrast, two-way service is more popular for better planned long trips that sometimes even require reservation ahead of time. At this point, we do not know the popularity of one-way or two-way service in the trailer-sharing in digital freight industry. We anticipate similar behavior, such that carriers working on less predictable local lanes with short lead-time and high demand volatility would favor this feature more as compared to stable regional lanes with long lead-time and low uncertainty. Therefore, it would be interesting to determine whether this feature is better for regional or local shipments.

## Chapter 3

### AUCTION PRICING MECHANISM FOR TRAILER-SHARING

#### **3.1 Introduction**

Etzion et al. [2006] showed that having a hybrid pricing mechanism that includes both dynamic and auction pricing models can increase revenue by enabling additional consumer segmentation. Combining of parallel auction and posted-price mechanisms allows the auction to be used to capture customers who were unable to purchase an item by the posted price through a dynamic pricing mechanism.

Priceline.com was the first online platform to implement this mechanism in 1999. The implementation of the hybrid pricing mechanism boosted sales by 82% in the third quarter of 2001 (Hann and Terwiesch [2003]). Other companies (e.g. Hotwire) have effectively introduced similar hybrid models (Amaldoss and Jain [2008], and Wang et al. [2009]). The drawback of hybrid models is it adds more complexity to setting a price for the payment management processes. It would be great to consider this pricing model for the initial price of a shipment job rather than ETR. In this thesis our focus is designing a pricing mechanism for an ETR job. Convoy has both pricing mechanisms, which it calls “Accept Now” and “Bid”, for dynamic and auction pricing, respectively. Therefore, in this chapter, we build an auction pricing mechanism and compare its performance with our dynamic pricing mechanism from the previous chapter.

Most current crowdsourcing platforms follow a dynamic pricing mechanism and offer a single posted price to workers. Workers must then decide if the offered price is worth their labor and either accept or reject the offered price. An alternate approach is to implement

an online procurement auction in which the platform allows workers to decide on the price through submitting their bids. In this type of pricing mechanism, workers can bid on the offered price based on their willingness to receive the payment, and the auctioneer can decide on the assignment and price to be paid to workers. We assume workers are rational. Therefore, the mechanism should be designed in a way that workers bid their true cost.

Designing an optimal auction pricing mechanism (which includes two steps: a winner of the auction and the corresponding payment) for assigning tasks to workers is crucial to online crowdsourcing platforms. In this chapter, we use a UCB-based algorithm (in combinatorial multi-armed bandit (CMAB))<sup>1</sup> for an online reverse second-price auction (special case of CMAB) for our monetary incentives pricing mechanism in which there is one buyer (Convoy) and multiple potential sellers (drivers) under budget constraints (the maximum amount that Convoy can pay a driver to do ETR).

Usually, second-price auction causes an overall reduction in bids, because it gives the bidders a chance to gain a little more than their original submitted offer (truthful incentives pricing mechanism).

In the combinatorial multi-armed bandit mechanism with rational arms, the player can select  $K < W$  arm candidates in each round and pulling each arm has a unique cost. After  $K$  arms are pulled, the player observes the random reward of each arm.<sup>2</sup> The rewards of arms are unknown prior. The goal of the player is to maximize the cumulative rewards under the budget constraint. We know each arm is rational and might misreport its cost to maximize its own payoff. Therefore, we use the combinatorial multi-armed bandit to ensure

---

<sup>1</sup>In contrast the classic multi-armed bandit models, combinatorial multi-armed bandit can select exactly  $K$  arms from  $W$  arm candidates in each round.

<sup>2</sup>The most complicated case is where pulling each arm has a unique cost (which is not in the scope of this study) and the player has a limited budget.

that each arm has no incentive to lie (false bid) <sup>3</sup>.

Therefore, in the general combinatorial MAB problem, the algorithm (player/platform) chooses the top K arms according to the decreasing order of the bang-per-buck ratio (UCB-based rewards to bids). Then, the algorithm determines the payment for each arm. The specific payment computation method is  $\frac{\hat{r}_i}{\hat{r}_{(K+1)}} \cdot b_{(K+1)}$ . This payment amount indicates that a winning arm that claims a bid greater than this payment will not win the auction process. However, a smaller bid will always win. After computing the payment for each winning arm, the player determines the total payment in this round. The arm selection and payment combination process will continue until the budget exhausts. <sup>4</sup>

This setup is popular in user selection in crowdsourcing platforms, Ad recommendation in social media, the rate allocation in wireless channels, etc. In our system, we consider the request of one driver for moving one trailer once. In the general case of our study, we can consider hauling a trailer by choosing multiple drivers for regional and local routes. <sup>5</sup>

We use a UCB-based algorithm for choosing an auction winner, because Convoy encounters the uncertainty of drivers' rewards. We select the UCB algorithm which is always optimistic about the uncertainty (each arm is a driver).

In the traditional multi-armed bandit problem, the player selects one arm at each time step and the assumption is that pulling every arm has the same cost. The model will be more complicated when arms (drivers) have their unique cost and the player (Convoy) has a limited budget (Tran-Thanh et al. [2012b], Amin et al. [2012b], Flajolet and Jaillet [2017]). Note that the budget is a strict constraint, and no workers can be selected if the budget is

---

<sup>3</sup>Unequal to its actual cost

<sup>4</sup>If the total payment in this round is greater than the remaining budget, the process will terminate.

<sup>5</sup>regional means between different cities while local means between different facilities in one city

exhausted, even if some tasks are left. We propose a reverse second-price auction pricing mechanism that uses the bang-per-buck ratio <sup>6</sup> (UCB-based reward to the corresponding bid). Our mechanism takes the submitted bids by drivers as inputs, selects a driver as winner by considering the bang-per-buck ratio selection criteria, and sets the payment (motivated by second-price auction) to the driver. Sometimes drivers do not trust the requester to reveal their true cost, or the cost may not even be known to the drivers. Communicating these true costs to the requester may be challenging in real-world settings. There is some research (Wu et al. [2019], Braverman et al. [2019], and Biswas et al. [2015]) that considers strategic workers (arms) in their model, who put their bid to maximize their payoff (the difference between the received payment and its cost). Therefore, we are interested in designing a truthful auction pricing mechanism in the presence of rational drivers to report their bids equal to their costs (the maximum payment the driver is willing to gain). Our proposed auction mechanism also ensures that the winning driver has a non-negative payoff due to individual rationality because otherwise, the driver will reject the offer. Similar to our previous UCB-based algorithm, we need to solve an optimization problem to get the near-optimal arm. We then analyze the regret of our auction pricing mechanism by simulation and compare the regret of the two proposed pricing algorithms. In our work, through theoretical analysis, we show that our auction pricing mechanism is truthful and individual rational <sup>7</sup>. Also, we demonstrate the computational efficiency of our mechanism through extensive simulation.

Our key contributions are: (1.) combining the auction pricing system into the UCB-based model, using bang-per-buck ratio, (2.) solving both winning arm selection and payment

---

<sup>6</sup>The bang-per-buck ratio of an arm  $i$  is given by the ratio of its value and cost. The key idea is to pretend that instead of budgets on resources we have costs for using them. We want to pick an arm which generates more reward at less cost. One natural way to formalize this intuition is to seek an arm which maximizes the bang-per-buck ratio. The issue is, we do not know the reward, and we really do not know the cost. Instead, we approximate the bang-per-buck ratio using the principle of optimism under uncertainty.

<sup>7</sup>Individual rationality means the pricing mechanism's arm selection needs to ensure that each strategic arm (driver) has a non-negative payoff.

computation problems, in the presence of rational drivers (arms), and (3.) executing a simulation to compare the performance of auction and dynamic pricing mechanisms. We found that auction pricing outperforms dynamic pricing.

### **3.2 Literature Review**

This section discusses the literature on a range of bidding models in operations research, computer science, and economics. A large body of research on crowdsourcing platforms has proposed various models for auction pricing in different areas, such as task assignment in an online environment (Buchbinder et al. [2007], Devanur and Hayes [2009b], Feldman et al. [2010], Devanur et al. [2011]), mobile crowdsourcing (Fan et al. [2015], Wei et al. [2015], Chen et al. [2016], Yang et al. [2012], and Xiao et al. [2016]), resource allocation in cloud computing (Nejad et al. [2014], Zhang et al. [2015], and Zhang et al. [2014]), and Internet advertising (Gonen and Pavlov [2007], and Babaioff et al. [2014]). The intersection of machine learning and designing pricing mechanisms is an active research area which includes work in different fields, including online mechanisms (Lavi and Nisan [2005], Maestre et al. [2018], Koolen et al. [2017], Shen et al. [2020], Drutsa [2020], Zhang et al. [2018], Khadge and Kulkarni [2016], Haliem et al. [2020]). Online reverse auctions models were pioneered by Van Tulder and Mol [2002] and Emiliani and Stec [2002]. Singer [2010] designed a budget feasible pricing mechanism to maximize a requester’s profit. Singer and Mittal [2013] designed a pricing framework that aims to maximize the number of performed jobs under a limited budget or to minimize the payments for a certain number of jobs. Bar-Yossef et al. [2002] and Lavi and Nisan [2004] introduced competitive online auctions for digital goods. Their results were improved by Kleinberg and Leighton [2003] (who implemented the Exp3 algorithm), Blum et al. [2004] (who implemented the Exp3 algorithm to optimize the competitive ratio), and Blum and Hartline [2005] (who implemented the Exp3 algorithm where the auction does not require foreknowledge of the range of bidders’ valuation) by using regret minimization algorithms. Similar to UCB, Exp3 attempts to place more weight on good arms that are promising. Exp3 behaves more like traditional machine-learning algorithms

than UCB, by learning weights to define how promising each arm is over time. UCB can effectively use the information from cross-learning despite it being drawn from a distribution that differs from the desired exploration distribution. Therefore, to fill out the gap of implementing the UCB algorithm in designing an auction pricing mechanism in on-demand platform systems, we designed our UCB-based bidding model by using the bang-per-buck ratio values. To the best of our knowledge, this is the first UCB-based algorithm (using bang-per-buck ratio) that addresses the budget constrained case and is designed in a way that selects both the winner of the auction and the corresponding payment.

There has also been growing interest in the incentive payment system for workers in on-line crowdsourcing platforms. Hajiaghayi et al. [2004] and Devanur and Hartline [2009] extended the bidding model under a limited budget constraint and the stochastic arrival of bidders. Babaioff et al. [2007] used the online knapsack secretary problem in designing a bidding model of budget constraint. Also, Babaioff et al. [2009] used the weighted secretary problem in their bidding model under budget. Singer and Mittal [2011] studied the bidding price model in the stochastic setting under a limited budget.

There is a series of papers that propose the procurement auction when under budget constraints for crowdsourcing (Chen et al. [2011], Singer [2012], and Bei et al. [2012]). The results of these papers can be categorized in two research streams: 1) when there is a general submodular utility function and, 2) when the pricing mechanism is based on the knapsack problem. The first line of research tries to optimize the utility function of the buyer under the constraint that the total payments made to the agents do not exceed a fixed budget. The second line of research casts the competitive bid optimization as an online knapsack problem to find an optimal competitive ratio where the knapsack has unlimited capacity. In contrast, our mechanism uses the ratios of UCB-based rewards to bids.

Nazerzadeh et al. [2008] built an auction pricing mechanism with approximate truthfulness and individual rationality properties. Similar to this setting (deterministic truthful MAB mechanism), Devanur and Kakade [2009] focused on maximizing the revenue of the mechanism. In our work, through theoretical analysis, we show that our auction pricing mechanism is truthful and individual rational.

### 3.3 Problem Setting and Algorithm

In this section, we design an auction-based UCB algorithm (MAB model). In our model, each arm is a driver and we assume our drivers are rational. Let us consider that there are  $W$  drivers,  $\mathcal{W} = \{1, 2, \dots, W\}$ , for a specific job in a defined auction time window. Here reward equals the net profit which comes from budget minus the payment plus adjusted value that comes from the extra parking slot charge. Parking price is different from time to time and its availability. Similar to other MAB algorithms, in our auction pricing mechanism, given  $W$  arms whose rewards are unknown in advance, Convoy selects one arm in each round, and its goal is to maximize the cumulative rewards over time by considering the limited budget.

In each time step  $t$ , we use a (non-negative) random variable  $r_i^t$  which denotes the reward of selecting arm  $i \in \mathcal{W}$ . Also,  $p_i^t(b_i)$  denotes the corresponding payment for the selected arm  $i$  with the bid  $b_i$  in round  $t$ . The reward for Convoy is equal to  $B - p_i(b_i) + v$ , where  $B$  is Convoy's budget and  $v$  is the value of dropping off an empty trailer to a facility, with deficit or surplus. If the facility has surplus <sup>8</sup>, then  $v > 0$ , otherwise  $v = 0$ . The  $v$  value comes from the extra parking space payment. These rewards are *i.i.d* with an unknown expectation,  $r_i = \mathbb{E}[r_i^t]$  (expected reward). Each driver has a cost  $c_i \in [\underline{c}, \bar{c}]$ , which is private information known only to the driver. In addition, each driver  $i$  submits a bid  $b_i \in [\underline{b}, \bar{b}]$  to

---

<sup>8</sup>Note that Convoy GOPlus is flexible to accept any request from drivers, even dropping off an empty trailer to a surplus facility. Therefore, due to unavailability of parking slots in the facility, Convoy needs to park its extra trailers somewhere else (depends on availability of parking) with some parking costs.

Convoy which does not necessarily equal the driver's cost.

At the beginning of the auction, each driver submits his bid to Convoy. In time step  $t$ , if Convoy selects a driver  $i \in \mathcal{W}$ , it can observe the reward ( $r_i^t$ ) at the end of time step  $t$ . This means Convoy must make a payment to the winning driver regardless of the reward being generated in the current round  $t$ . The designed mechanism should ensure the payment amount is at least equal to the bid of the winner in the auction, by considering the limited budget  $B$  to maximize Convoy's reward (profit). Also, in the presence of rational drivers, the mechanism should warrant that drivers have no incentive to lie.

Our auction pricing mechanism includes two main processes: (1.) how Convoy should choose the winning driver at each time step in order to maximize the expected reward under the budget constraint; (2.) at each time step, how Convoy should determine the payment for the winning driver to keep the algorithm truthful.

To design an arm-pulling policy to maximize the reward, we need to manage the trade-off between choosing the best arm based on the sampling (exploitation) and trying a sub-optimal arm to find the potentially optimal arm (exploration), we used the bang-per-buck ratio. This ratio is defined as the UCB-based reward to the bid. The principle behind UCB is optimism in the face of uncertainty.

Let  $N_i^t$  be the number of times arm  $i$  is chosen, and  $\bar{r}_i^t$  be the average empirical reward of arm  $i$ , until time step  $t$ . These values will be updated at the end of each time step as follows:

$$\left\{ \begin{array}{l} N_i^t = N_i^{t-1} + 1 \\ \bar{r}_i^t = \frac{\bar{r}_i^{t-1} \cdot N_i^{t-1} + r_i^t}{N_i^{t-1} + 1} \end{array} \right.$$

Now, let us define  $\hat{r}_i^t$  as the UCB-based reward for arm  $i$  at time step  $t$  as follows:

$$\hat{r}_i^t = \bar{r}_i^t + \sqrt{\frac{2 \ln(t)}{N_i^t}}$$

In the first time step, the algorithm chooses all arms (drivers) to obtain and initialize the parameters  $N_i^t$ ,  $\bar{r}_i^t$ , and  $\hat{r}_i^t$ . Then, at each time step, the algorithm will choose arm  $i$  by considering the limited budget constraint. Therefore, at the beginning of each time step  $t$ , it can obtain  $\hat{r}_i^{t-1}$  for arm  $i$  and initialize the chosen arm and the corresponding payment. Now, the algorithm sorts all arms based on the bang-per-buck ratio ( $\frac{\hat{r}_i^{(t-1)}}{b_i^t}$ ), in decreasing order. Then, the algorithm chooses the arm with the highest bang-per-buck ratio. Then, in the same step  $t$ , the algorithm calculates the suitable payment for the chosen arm (winning driver) based on the second lowest bid (second-price auction idea) as follows (see Theorem 3.1):

$$p_i^t(b_i^t) = \frac{\hat{r}_i^{t-1}}{\hat{r}_{i'}^{(t-1)}} \cdot b_{i'}^t$$

where  $i'$  is the second lowest bid.

After calculating the payment, the algorithm compares the payment at the current time step with the available budget. If  $p_i^t > B^{(t-1)}$  then the algorithm will stop; otherwise, the algorithm observes the reward in the current time step and updates the parameters and the budget.

### 3.4 Results

In this section, we discuss the truthfulness, individual rationality, and computational efficiency of our auction-based UCB algorithm.

#### 3.4.1 Proof of Truthfulness

**Theorem 3.1.** *In each round, UCB-based auction pricing is truthful.*

*Proof.* An auction pricing mechanism is truthful if and only if it satisfies the following two conditions: [Papadimitriou and Singer [2010], Jia et al. [2009]]

1. The allocation rule or the winning arm selection process is monotonic. This means that if driver  $w_i$  (arm  $i$ ) wins the auction by claiming bid  $b_i$ , he also wins by placing a bid such that  $b'_i \leq b_i$ .
2. Each winning driver or winning arm is paid the calculated price as a threshold. This means that if the driver bids higher than this price, he will not win the auction.

First, we need to prove that at each time step, our winning arm selection is monotonic.

Based on the arm selection criteria,  $\frac{\hat{r}_i}{b_i} \geq \dots \geq \frac{\hat{r}_k}{b_k} \geq \frac{\hat{r}_{k+1}}{b_{k+1}} \geq \dots \geq \frac{\hat{r}_W}{b_W}$ , if the driver  $w_i$  can win the auction with  $b_i$ , the driver has to win when submits a smaller bid  $b'_i = b_i - \eta$ ,  $\eta > 0$  (based on the greedy selection  $\frac{\hat{r}_i}{b_i}$ ).

Now, we need to prove the second condition. In general, second-price auction is always a truthful mechanism (Jia et al. [2009]).

Again, based on the selection criteria ( $\frac{\hat{r}_1}{b_1} \geq \dots \geq \frac{\hat{r}_k}{b_k} \geq \frac{\hat{r}_{k+1}}{b_{k+1}} \geq \dots \geq \frac{\hat{r}_W}{b_W}$ ), the threshold price payment for the winning driver with bid  $b_i$  is calculated by  $p_i(b_i) = \frac{\hat{r}_i}{\hat{r}_{i'}} b_{i'}$ , where  $i'$  is the second lowest bid. Therefore, if the bid by the driver  $i$  is lower than  $p_i(b_i)$ ,  $b_i \leq p_i(b_i)$ , then  $\frac{\hat{r}_i}{b_i}$  has to be greater than  $\frac{\hat{r}_{i'}}{b_{i'}}$ . Therefore, based on the selection criteria, this means that the bid  $b_i$  has to be chosen before  $b_{i'}$ . On the other hand, let's consider the case if the bid  $b_i$  is greater than the threshold or calculated price,  $b_i > p_i(b_i)$ , then the bid  $b_{i'}$  has to be chosen before  $b_i$  as a winning arm. This means that the driver  $i$  will lose the auction.

Therefore, by considering this condition together with the first condition, we can conclude that  $p_i(b_i) = \frac{\hat{r}_i}{\hat{r}_{i'}} b_{i'}$  is the threshold payment. The higher bid does not increase the payment

(due to potentially exceeding the threshold price), and it might increase the chance of failure based on the payment formula. Thus, the auction algorithm can ensure the truthfulness of all rational drivers (arms) at each time step.  $\square$

### 3.4.2 Proof of Individual Rationality

**Theorem 3.2.** *In each round, UCB-based auction pricing is individual rational<sup>9</sup>.*

*Proof.* If driver (arm)  $i$  with corresponding bid  $b_i$  loses the auction in round  $t$ , his payoff will be 0, otherwise  $p_i^t(b_i^t) - c_i^t$ .

The auction mechanism payment at time  $t$  is defined as follows:

$$p_i^t(b_i^t) = \frac{\hat{r}_i^{(t-1)}}{\hat{r}_{i'}^{(t-1)}} b_{i'}^t$$

where  $i'$  is the second lowest bid.

Here, in time step  $t$ , because a driver with bid  $b_i$  won with respect to a driver with bid  $b_{i'}$ , we have,  $\frac{\hat{r}_i^{(t-1)}}{b_i} \geq \frac{\hat{r}_{i'}^{(t-1)}}{b_{i'}}$ . Therefore, based on the arm selection criteria, we have:

$$\begin{aligned} b_i^t &\leq \frac{\hat{r}_i^{(t-1)}}{\hat{r}_{i'}^{(t-1)}} \cdot b_{i'}^t \\ &= p_i^t(b_i^t). \end{aligned}$$

Also, as we proved the truthfulness of our auction pricing mechanism above, for all arms,

---

<sup>9</sup>Individual rationality means the pricing mechanism's arm selection needs to ensure that each strategic arm (driver) has a non-negative payoff.

we have  $c_i = b_i$ , where  $c_i$  is the private cost of a driver, and  $b_i$  is his bid. Therefore,

$$\begin{aligned} c_i^t = b_i^t &\leq \frac{\hat{r}_i^{(t-1)}}{\hat{r}_{i'}^{(t-1)}} \cdot b_{i'}^t \\ &= p_i^t(b_i^t). \end{aligned}$$

This means, in any time step  $t$ ,

$$p_i^t(b_i^t) - c_i^t \geq 0$$

Therefore, at each time step  $t$ , the winner's payoff is positive, which means the individual rationality holds.  $\square$

### 3.4.3 Proof of Computational Efficiency

To demonstrate computational efficiency we ran the following simulation. We set the number of arms,  $W$ , from the set  $\{50, 60, 70, 80, 90, 100\}$ , for a different budget from the set  $\{6 \times 10^4, 8 \times 10^4, 10 \times 10^4, 12 \times 10^4, 14 \times 10^4, 16 \times 10^4, 18 \times 10^4\}$ . Then, we generate the rewards for each arm by setting the expected reward,  $\mathbb{E}[r_i]$  from  $[0.1, 1]$ . We choose the Gaussian distribution for generating reward values. Also, we use the uniform distribution for generating the bid of drivers (generating  $b_i$  randomly from the range  $[0.1, 1]$ ). Figure 3.1 shows the total rounds when the number of arms and budget are changed from 50 to 100 and  $6 \times 10^4$  to  $18 \times 10^4$ , respectively. As you can see, the highest running time is about 250 seconds, when the number of drivers is 100 and the available budget is  $18 \times 10^4$ . This is important because Convoy must process the payment computation in nearly real time, so that the total reward can be maximized under the budget constraint.

Because of the lack of data issue, in empirical results usually we see the performance of the algorithms based on the total reward or regret which is the goal of the MAB algorithms. Figure 3.2 shows that by increasing the budget (given a fixed number of drivers), the total reward also increases. This is because the scale of increasing the budget is in the

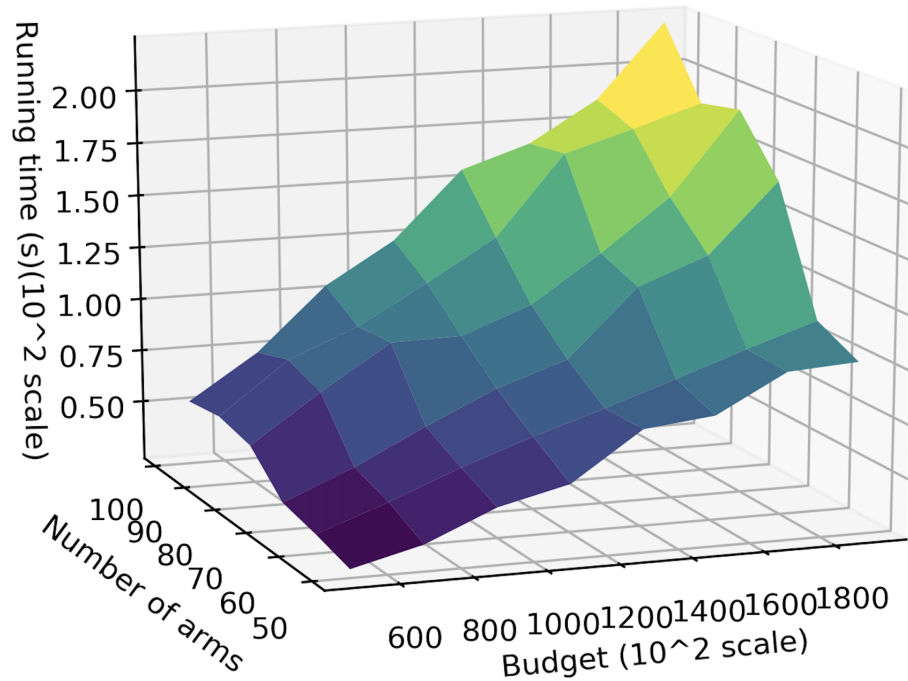


Figure 3.1: Computational Efficiency

order of  $10^2$ , but the increase of the bid value is in the order of 0.01. The scale difference between budget and bid occurs due to as Convoy increases its budget, it charges shippers more because of their rush in shipment or an unwanted lane. Also, in general, with more budget, more jobs can be finished on time.

To analyze the truthfulness of our auction pricing mechanism through the simulation, we randomly chose an arm and changed the corresponding bid (given all other settings are fixed). As long as the bid is lower than the payment, the arm has the potential to win if its bid is still lower than other arms whose bids are below the payment. Also, if the arm wins, its payoff remains the same because the payoff will not be changed based on the variation in the arm's bid. However, if the bid goes above the payment, the arm will lose the auction and not receive a payoff. Therefore, this guarantees that each arm has no incentive to lie about its true cost in our UCB-based auction mechanism.

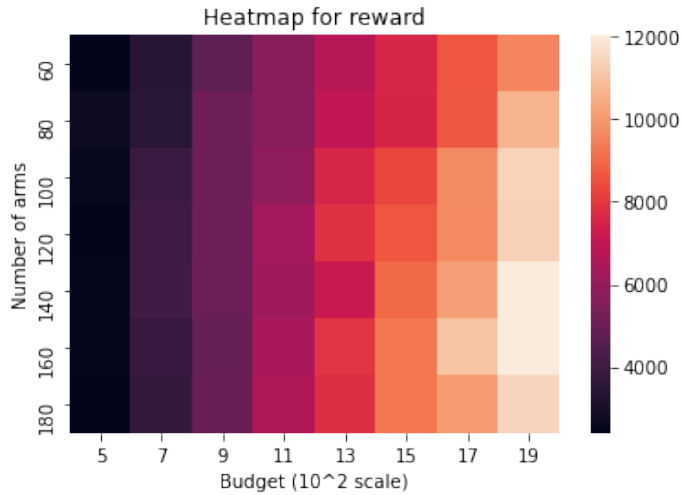


Figure 3.2: Total Reward vs. Number of arms and Budget

Besides showing the truthfulness through the simulation, we show the individual rationality of our auction mechanism as well. To do this, in each time step, we stored the payment amount for a winning arm and the number of times it won the auction during the run time. Then, we computed the average payment for each arm, and we saw that the average payment for each arm is greater than or equal to its bid and cost (see Figure 3.3). This means our auction pricing mechanism is also individual rational. One of the components in the high-level decisions associated with the revenue model is the pricing mechanism. The most common and simple mechanism is dynamic pricing, because it is easy to communicate with customers and enables an immediate transaction. On the other hand, in the presence of limited information about the customers' preferences, the firm might face a big challenge in determining the best price. An alternative pricing mechanism is auction pricing. As we discussed in this chapter, a well-designed auction mechanism can capture the true cost (bid) of customers. This benefits the firm since the final offered payment (price) reflects the true cost of customers. However, compared to dynamic pricing, auction pricing enforces more transactional costs, since it needs to wait to gather and monitor all bids. There is

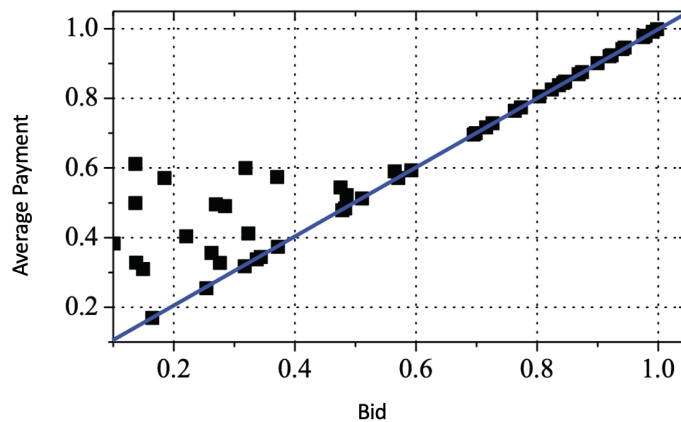


Figure 3.3: Individual Rationality of Auction Pricing Mechanism

also the uncertainty of completing the transaction. Therefore, given the limitations of both dynamic and auction pricing, one solution is to offer both of them to customers (hybrid mechanism). In fact, in this way, we can consider additional customer segments to increase revenue (Etzion et al. [2006], Wang et al. [2009]).

The following figure (Figure 3.4) displays the performance (average regret) of both dynamic and auction pricing mechanisms with an increasing budget, the former of which is reduced as the budget increases. This figure shows that the performance of our auction pricing mechanism is better than our dynamic pricing mechanism, given the same budget. Since we have a wide range of preferences<sup>10</sup>.

---

<sup>10</sup>In the presence of sufficiently dispersed customer preferences, we expect the auction mechanism outperform the dynamic pricing mechanism (Wang [1993]).

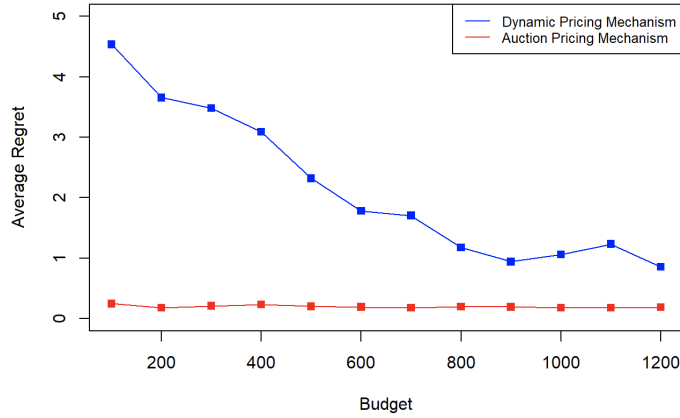


Figure 3.4: Comparing Dynamic and Auction Pricing Mechanisms Performance

### 3.5 Conclusions and Future Work

We started with a mechanism for the dynamic pricing and then extended it to our auction pricing mechanism. We proved that our bidding model is empirically efficient compared to the dynamic pricing mechanism and dramatically outperforms it. We also proved the truthfulness and individual rationality of our auction pricing mechanism. Our experiments support the practical applicability of our mechanisms on crowdsourcing platforms to increase their efficiency.

It would be useful to extend our approach to design mechanisms which can take into account different utility functions for both Convoy and the drivers on the platform. It would be interesting to show their theoretical guarantees. In this work, we assumed a large pool of drivers. Perhaps a more natural approach is to assume a limited number of drivers, as well as a limited amount of time.

## Chapter 4

**PERFORMANCE EVALUATION OF CLINICAL ENTITIES  
EXTRACTION: AMAZON COMPREHEND MEDICAL VERSUS  
MANUAL DATA EXTRACTION****4.1 Introduction**

A big limitation of ongoing clinical research, particularly retrospective studies, is its reliance on long manual chart reviews to collect data. This consumes the bulk of a project, as the statistical analysis can usually be performed quickly in comparison. Often these records are unstructured, even with the use of an Electronic Medical Record (EMR) system. EMRs are designed to aid with billing and documentation. These records also act as a log for physicians to recall their medical decisions, but they are not designed with the intent of collecting properly formatted data for research, leading to the significant time commitment needed for manual data extraction. In the context of this manuscript, structured data is a tabular dataset that has well-defined columns and is easily searched, sorted, and used in statistical analysis, while anything else is unstructured. Aside from the time commitment presented by unstructured data, other significant barriers to using this data in research include inconsistencies in reviewers' decision-making and across records as patients change between care centers.

Researchers have been working to address these shortcomings by expanding the scope of their data sources or integrating machine learning and computerized automation into the extraction process. This second method is part of a fast-growing branch of data science called Natural Language Processing (NLP) and involves automating the processing of human language. This can take many different forms, ranging from sentiment analysis, which is the analysis of text to extract subjective information, to Named Entity Recognition

(NER), which is the identification and classifications of entities in unstructured text, and even predictive modeling. NLP has proven to be crucial in lessening the time investment needed to construct a dataset. Ideally, an algorithm would be applicable to any type of medical records and would extract all relevant data needed for the study without need for human intervention. If perfected, this would reduce the data collection phase to a small fraction of what it is today and would provide an automated solution to structure the data. Unfortunately, this does not yet exist, but NLP has been successfully used in tandem with manual reviews to enhance the process. NLP has been applied in a wide range of clinical applications, such as being used to scrape published literature to explore connections between cancer and certain genes, extract relevant data from clinical records, and to predict clinical outcomes. These capabilities show, with more development, algorithms will be able to contribute more extensively, and eventually take over the data extraction process.

The use case that interests us is the application of NLP, more specifically in the form of NER, to automate data extraction from clinical records. New tools are being developed to speed up manual extraction by using NER, which when properly trained, allows a model to automatically identify and extract relevant data from a body of text. Amazon has recognized the importance of clinical text mining and developed a software add-on, Amazon Comprehend Medical (ACM), to Amazon Web Service, which provides clinical text processing. In this study, we intended to evaluate the effectiveness of using Amazon Comprehend Medical in evaluating a cohort of patients with multiple myeloma seen at the Seattle Cancer Care Alliance (SCCA). This tool uses NER to extract relevant data from unstructured texts and compile them into datasets. More specifically, it recognizes a feature in a document, extracts it, and classifies it as a drug, symptom, diagnosis, or another appropriate classification. It also records the location of the feature in the clinical records. We compared the NER features to manually extracted data to explore how the two sources complement each other and to evaluate the accuracy of NLP.

We decided to focus on the applications of ACM on multiple myeloma for this study. Multiple myeloma is a form of cancer of bone marrow plasma cells with no known cure. In the US, there were around 35,000 new cases in 2021 with a 5-year relative survival rate of 55%<sup>1</sup> between 2011 – 2017, putting it at the 14th most common cancer<sup>2</sup>. Researchers are still studying the most effective treatments to treat it, and current treatments find some success in limiting the spread of the disease and sometimes cause complete remission. We chose this disease because the relatively long survival time of patients, along with patients' regular monthly follow-ups, have allowed us to collect a large dataset to work with and analyze. In addition, the SCCA is one of the leading bone marrow transplant centers in the country, meaning there is a wealth of data to mine.

#### *4.1.1 Patients Information*

This study uses data from 831 patients who were treated for multiple myeloma at the SCCA and University of Washington affiliated hospitals between 2003 and 2018 (see Table 4.1). The data was collected from clinic notes, pathology reports, and radiology reports recorded throughout the patients' treatment and diagnostic histories. The data was collected in two ways: manual and automated extraction using Amazon Comprehend Medical. Specifically, datasets on Fluorescence in situ hybridization (FISH<sup>3</sup>) results, bony lesions, lines of treatment, plasma cells, radiation therapy, staging systems, survival information, and demographics were collected manually, while diagnoses, labs, medications, signs, symptoms, and billing codes were collected using ACM. For results from ACM, only features with no negation score were considered, so that we may be certain that all the features occurred.

---

<sup>1</sup>The 5-year survival rate tells you how many of people (out of 100) live at least 5 years after the cancer is found.

<sup>2</sup>[www.seer.cancer.gov](http://www.seer.cancer.gov)

<sup>3</sup>FISH is a laboratory-based test that looks for gene changes in cells. FISH test looks for specific genes or parts of genes and helps build out the full picture of a cancer diagnosis by zooming in on the genetic material in the cell.

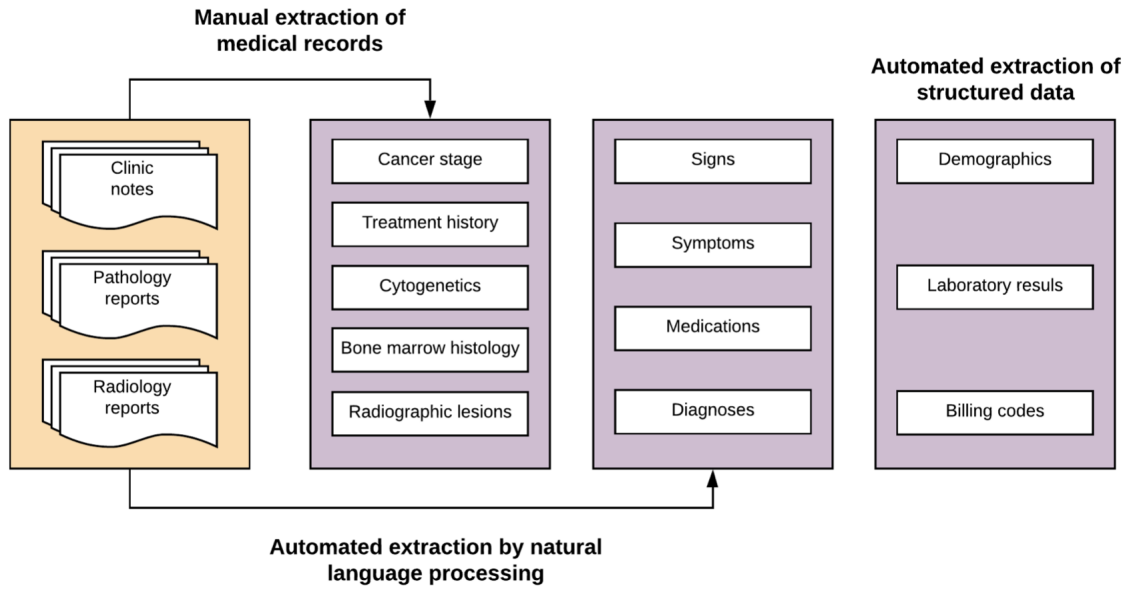


Figure 4.1: Data Extraction

Table 4.1: Summary of Patients Characteristics

Data Set	Categories	Overall(N=831)
Age at Diagnose	Median	59
	[Min, Max]	[22, 82]
Patient Sex	Female	330 (39.7%)
	Male	501 (60.3%)
Racial Group	White	716 (86.2%)
	Black	34 (4.1%)
	Asian	23 (2.8%)
	Other	18 (2.2%)
	Not Reported	40 (4.8%)
Status	Dead	271 (32.6%)
	Alive	130 (15.6%)

	Unknown	430 (51.7%)
International Staging System (ISS) Stage	I	148 (17.8%)
	II	174 (21%)
	III	162 (19.5%)
	R-ISS Staging System	46 (5.5%)
	Missing	301 (36.2%)
Fluorescence in situ hybridization (FISH)	del13q	190
	1q+	157
	t(11:14)	143
	del17p	123
	t(4:14)	78
	del1p	39
	t(14:16)	24
	t(6:14)	1
	Other	568
Treatment Duration (Days)	0-1000	181 (21.8%)
	1001-2000	286 (34.4%)
	2001-3000	167 (20.1%)
	3000+	180 (21.7%)
	Unknown	17 (2%)

Of these 831 patients, the median age at diagnosis was 59 and 60% were male, 86% were white, and 4.1% were black. Roughly twice as many patients died than lived, based on the follow up data available. The patients have a median survival of 4.92 years after their day of diagnosis with multiple myeloma. Of the automatically extracted records, patients were found to have had an average of 8 diagnoses, 2 billing codes, 10 signs, 8 symptoms, 1 line of treatment, and 8 medications recorded per day of follow up.

Table 4.2: Summary of Datasets

Data	No. of Events	Events/Patient	Events/Patient/Day
Diagnoses	674,388	811	8
Billing Codes	360,053	433	2
Signs	597,569	719	10
Symptoms	650,426	782	8
Treatments	9,775	11	1
Medications	647,778	779	8

#### 4.1.2 *Natural language processing*

Investigators at the Fred Hutchinson Cancer Research Center used Amazon Comprehend Medical to perform entity extraction from the study population. This data has been linked to the database of manually extracted features in addition to available structured clinical data. To protect private health information, a de-identified version of this database has also been developed to facilitate analyses with colleagues who have not undergone HIPAA <sup>4</sup> training. Below are some summary figures for the Natural Language processed dataset:

---

<sup>4</sup>Health Insurance Portability and Accountability Act

Table 4.3: Summary for the Natural Language Processed Dataset

Variables	Value
Number of patients	831
Number of encounters	212,064
Median duration between encounters (years)	2.2
Median number encounters per patient	193
Number of note types	454
Median number note types per patient	43
Number of entities	26,733,994
Median number entities per patient	23,550
Earliest encounter	November 22, 2003
Latest encounter	October 8, 2018

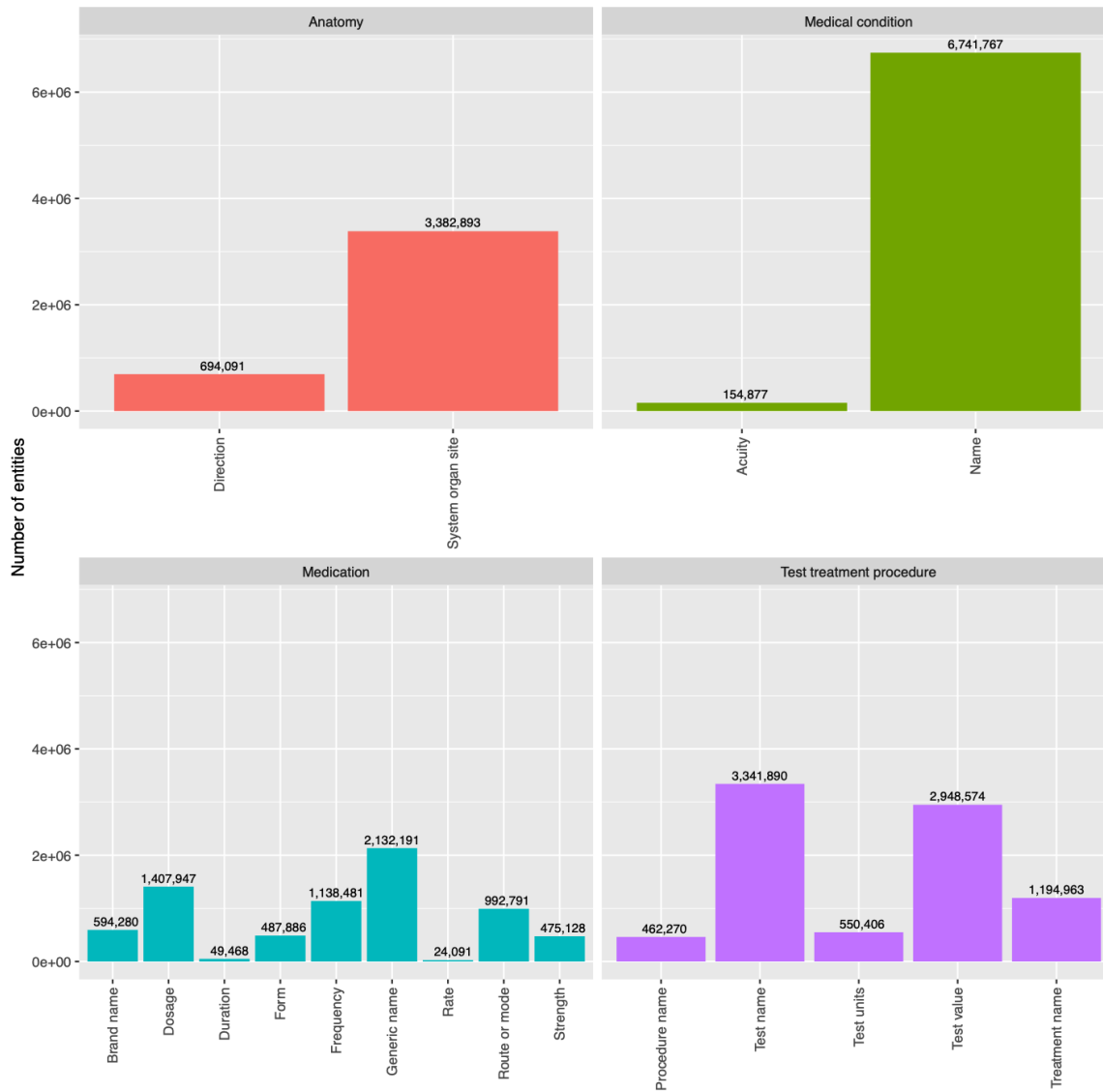


Figure 4.2: Number of entities in different datasets extracted through NLP

## 4.2 Methods

### 4.2.1 Bidirectional Long Short-Term Memory

ACM is driven by a state-of-the-art deep learning algorithm, which utilizes a neural network framework with Bidirectional Long Short-Term Memory (BiLSTM) networks for negation scope detection. These LSTM encoders are at character and word level. Currently, ACM

supports clinical notes written in English.

#### *4.2.2 Data Analysis*

Extracting medical information manually through unstructured clinical text notes can be time-consuming and result in errors. Therefore, there is need for a human-free system to automatically extract structured medical information from unstructured notes. The objective of this study is evaluating an automated extraction via ACM compared with human (manual) data extraction. We select a population of 831 patients diagnosed with multiple myeloma who received treatment at the Seattle Cancer Care Alliance from November 2003 to October 2018. Each patient provides prior consent allowing researchers access to their clinical data. All patients underwent first-pass manual abstraction of 47 features known to be clinically relevant to the disease. These include features such as cancer stage, pathology, radiology, and chemotherapy history. In addition, laboratory data, demographics, and billing codes are available in a structured database for each patient. We want to answer the following question through data analysis:

Can Amazon Comprehend Medical extract entities from clinical notes and identify medical information better than human?

Many research questions can be answered with descriptive analysis. Therefore, graphing the data distribution is an essential step in the exploratory analysis of quantitative data. Furthermore, the data distribution examination is fundamental for both multivariate regression and inferential analyses. An overview of our datasets is illustrated in Figures 4.3 and 4.4.

Figure 4.3 represents the distribution of patients' age at the admission time. It also illustrates the frequency of male and female patients in our data (number of females are  $\sim 65\%$  of the number of males) and the distribution of different race groups among our patients

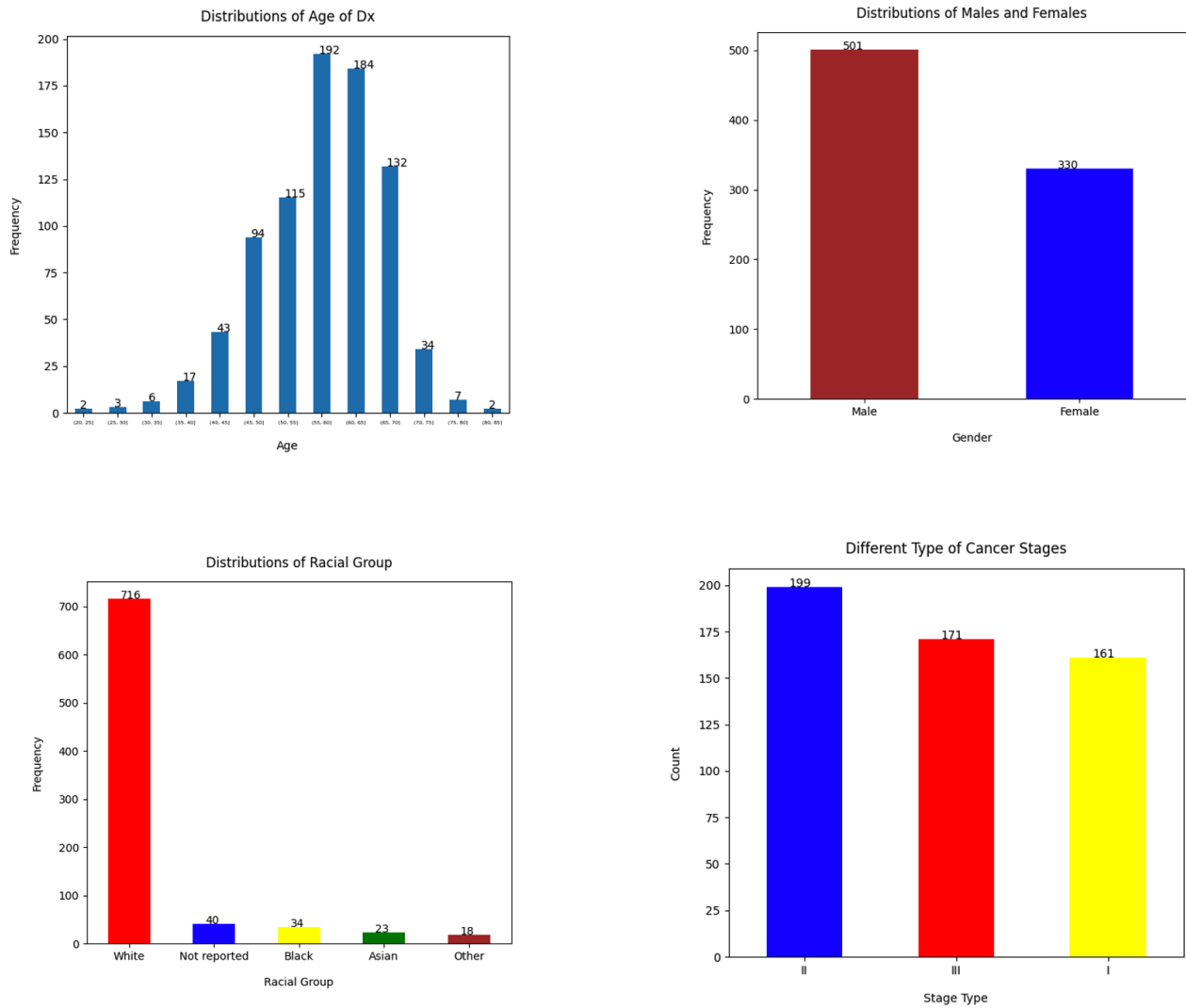


Figure 4.3: Distribution Information from Manual Extraction

(race white constitutes the majority of our sample data). Figure 4.3 depicts the frequency of varying cancer stages among our patients (stage II is the most frequent one). Figure 4.4 illustrates the top 50 terms in medication, symptoms, diagnoses, and signs, respectively.

The top 50 terms in the above figure were determined after the following processes:

1. Removing all extracted data after the start day of the last phase of therapy

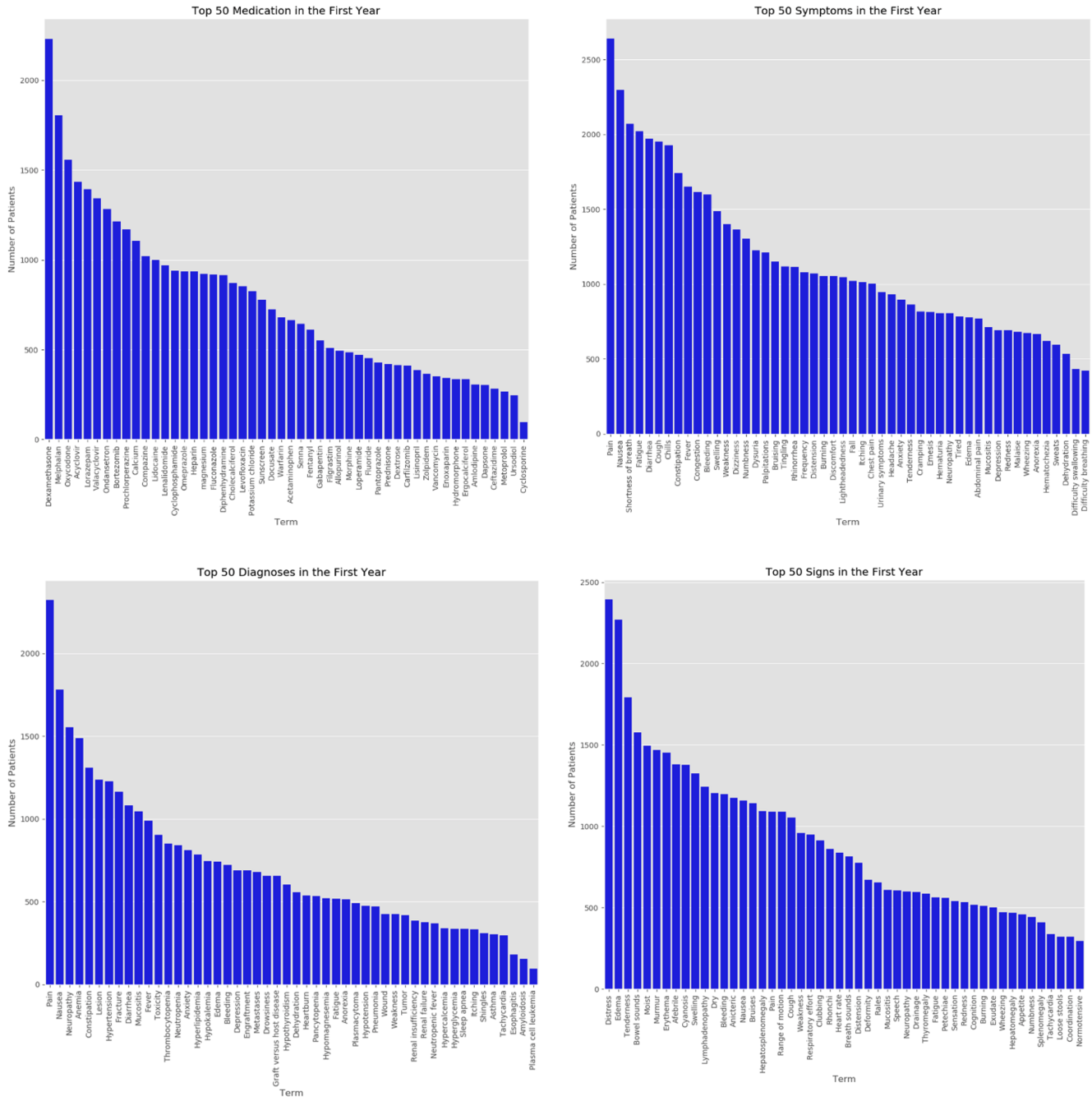


Figure 4.4: Distribution Information from NLP

2. Deleting all terms with non-null negation
3. Deleting all terms with a F-score less than 0.9
4. Removing duplicate terms for each patient per visit
5. Counting unique terms

Choosing data collection methods and identifying ways to analyze and interpret the collected data is an arduous process, but early-stage data exploration can give critical guidelines for later analysis. It is common for researchers to start their projects with a preliminary analysis, including quantitative, qualitative, visualizing, or any combinations, before choosing a method. We investigated the use of several different techniques for quantitative, qualitative, and mixed data, including graphic representations.

#### *4.2.3 Quantitative Analysis Techniques*

In this section, we discuss three techniques for quantitative analysis: 1) inferential, which is used for conclusions about the population, 2) descriptive (such as studying on collecting qualitative data in textual form), and 3) associational (which is the use of statistics to analyze any complex relations among variables). Also, by using inferential statistical techniques, such as the multimodal approach on our numerical data, we generalized the outcomes from our probability samples for the population. The descriptive statistics depict the actual data to explore the shape of data, while the inferential statistics estimate probable value. In associational techniques, we used correlation and regression methods to interpret the complex relations among variables. This approach uses maximum likelihood estimation. Although it is more computationally expensive, it is highly beneficial to investigate how well the data fit models.

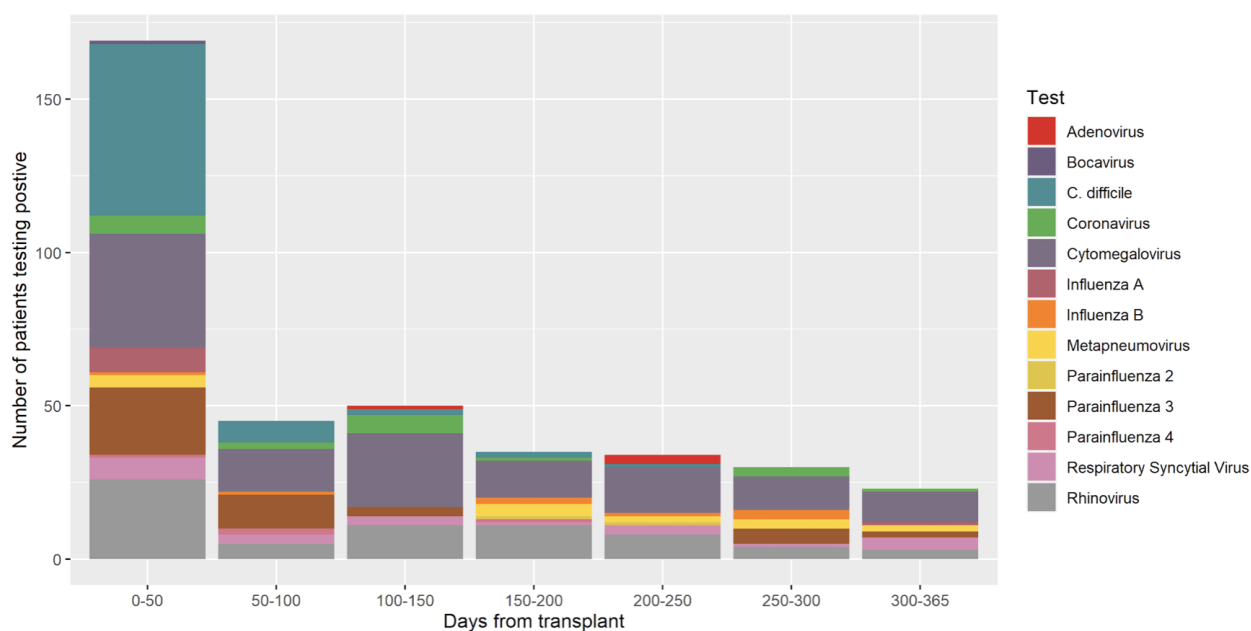


Figure 4.5: Distribution of Infections

#### 4.2.4 Statistical Analysis

We used major statistical analysis packages in open-source software, R and Python. We consider two research questions.

The first research question is determining if clinical features recorded up to 3 months prior to autologous stem cell transplant are predictive of infection within a year post-transplant among patients with multiple myeloma. We first ran some statistical tests, including fisher's tests and Bonferroni adjustments (see Figures 4.5 and 4.6).

The second research question is predicting the time-to-next-treatment after the first relapse phase. To predict the time-to-next-treatment (TTNT), first we apply some feature selection methods. Then, we split the data into 70% train and 30% test data. Some machine learning algorithms, including xgboost, stepwise, jackknife, linear, ridge, lasso, elastic net,

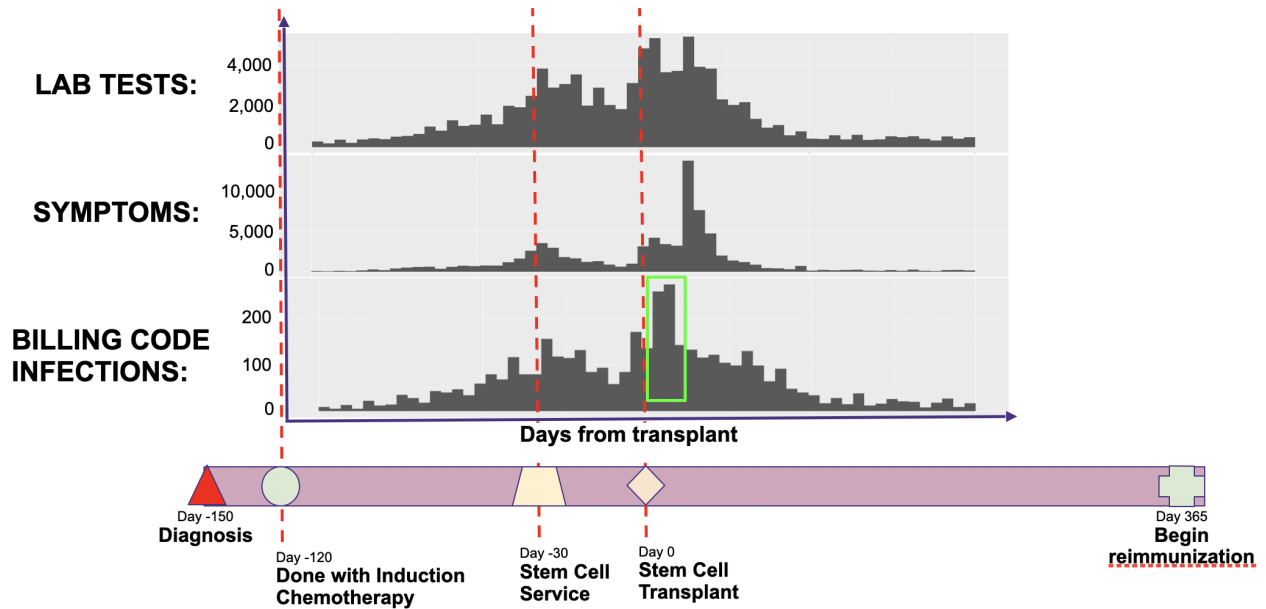


Figure 4.6: Feature distribution for infected patients

Bayesian, and polynomial regression, are applied to the data set. All models were evaluated on the same test dataset by the following error measures: 1) Root mean squared error (RMSE), 2) Mean absolute percentage error (MAPE), and 3) R-squared. Regression model coefficient and models' performance are visualized in Figures 4.7 and 4.8.

#### 4.2.5 Descriptive Analysis

Because the patients' data is confidential and not available to readers, we provide some basic descriptive statistics to show our data, features, and patterns. Descriptive analysis has a central role in identifying both the problems and patterns in missing data. It is also a crucial step for researchers to do further complicated analysis. Unlike multivariate statistics, descriptive statistics are used to investigate only one variable at a time, which means they do not identify the relations among the variables. Besides exploring the distributions of the data, descriptive statistics summarize the principal features of the data. Here are the categories of descriptive analysis we report in our research:



Figure 4.7: Predicting TTNT results

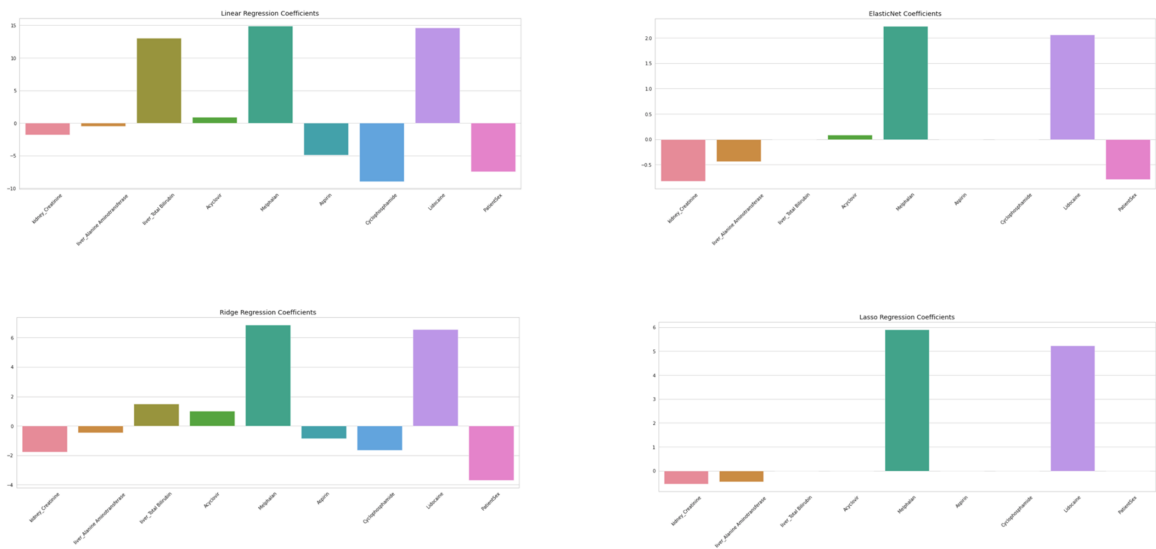


Figure 4.8: Regression model coefficients visualization

1. Measures of Central Tendency: mean, median, and mode
2. Measures of Dispersion: variance and standard deviation
3. Measures of Relative Position: percentile and the z-score
4. Measures of Association: correlation
5. Measures of Margins of Error: standard error and confidence interval

In descriptive statistics, we try to provide sufficient information to readers about who and what was studied, how and how many entities were selected, and what were their characteristics and patterns. Also, it can help other interested researchers to use the results in their meta-analyses.

#### *4.2.6 Outliers*

Outliers significantly influence the outcome of any analysis. Usually, most of the outliers are mistakes in the extraction process. Therefore, it is vital to identify the outliers at the beginning of starting the research and delete them before running any analyses such as mean, variance, standard deviation, t-test, correlations, and regression coefficients.

Figure 4.9 shows one of the outlier analyses we did before predicting the TTNT with the following predictors: kidney function, liver function, medications, and sex.

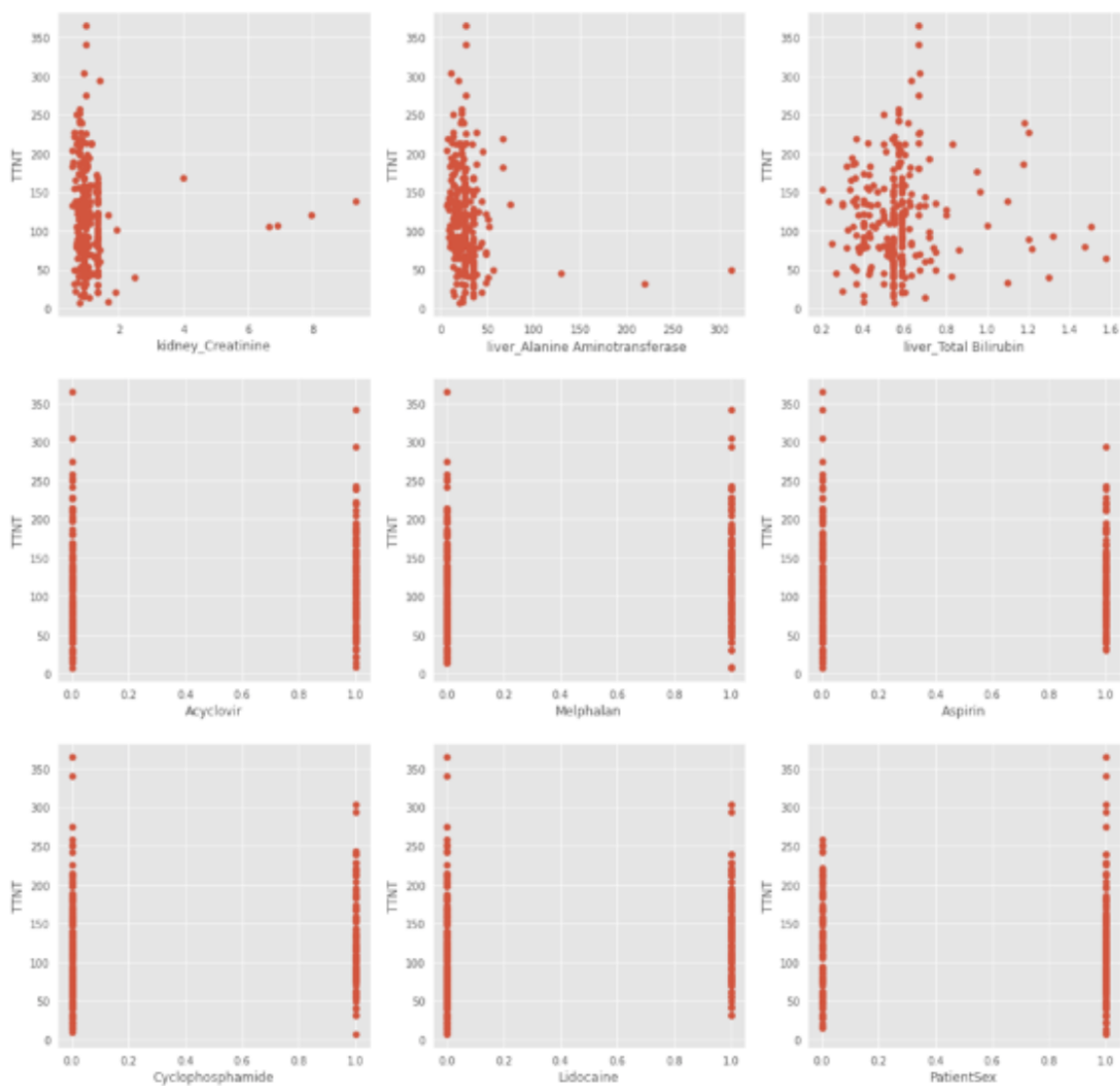


Figure 4.9: Dataset scatterplots by the dependent variable

Below we describe a summary of applying different outlier removal methods and their effect on the model performance:

#### Local Outlier Factor on the Entire Dataset

- Removed  $\sim 70$  rows, from 200 to 130 rows.

- Slight improvement in stepwise, ElasticNet, and Bayesian models. Decrease in performance for all other models.
- Below: left table is no outlier removal; right table is with outlier removal.

	Model Name	RMSE	MAPE	R2		Model Name	RMSE	MAPE	R2
0	xgboost	53.677474	0.641035	0.231580	0	xgboost	56.088013	0.669923	0.161014
1	stepwise	58.254232	0.773154	0.094956	1	stepwise	58.100706	0.706575	0.099720
2	jackknife	58.347698	0.784306	0.092049	2	linear	59.128475	0.665623	0.067587
3	linear	58.347729	0.784303	0.092048	3	jackknife	59.132928	0.665678	0.067447
4	ridge	59.436676	0.794433	0.057842	4	ridge	59.477611	0.690625	0.056544
5	lasso	59.756936	0.795793	0.047661	5	lasso	60.381047	0.682492	0.027665
6	elasticnet	60.775126	0.809960	0.014931	6	elasticnet	60.593035	0.705547	0.020825
7	bayesian	61.583675	0.818436	-0.011454	7	bayesian	60.743155	0.711214	0.015967
8	polynomial	79.204878	0.879420	-0.673087	8	polynomial	160.142307	1.227314	-5.839541

Figure 4.10: Performance of Local Outlier Factor Removal Method

### Brute-Force Removal of Outliers in Kidney and Liver Features

- Removed all rows with Kidney  $> 2.5$  or Liver  $> 60$ . Led to the removal of 10 rows, from 200 to 190 rows.
- Interestingly, there is a huge decrease in performance by removing these ten rows. All model parameters were fine-tuned on this new data.
- Below: left table is with no outlier removal; right table is with outlier removal.

	Model Name	RMSE	MAPE	R2		Model Name	RMSE	MAPE	R2
0	xgboost	53.677474	0.641035	0.231580	0	xgboost	68.063580	0.706298	0.090044
1	stepwise	58.254232	0.773154	0.094956	1	stepwise	68.115504	0.674236	0.088655
2	jackknife	58.347698	0.784306	0.092049	2	ridge	68.659658	0.705675	0.074036
3	linear	58.347729	0.784303	0.092048	3	elasticnet	68.719434	0.711893	0.072423
4	ridge	59.436676	0.794433	0.057842	4	jackknife	68.720193	0.711973	0.072403
5	lasso	59.756936	0.795793	0.047661	5	linear	68.720516	0.711967	0.072394
6	elasticnet	60.775126	0.809960	0.014931	6	lasso	68.999544	0.686143	0.064846
7	bayesian	61.583675	0.818436	-0.011454	7	bayesian	69.328460	0.697937	0.055909
8	polynomial	79.204878	0.879420	-0.673087	8	polynomial	144.121963	1.234822	-3.079915

Figure 4.11: Performance of Brute-Force Outlier Removal Method

### Mark Outlier Values for Kidney and Liver Features as Missing, then Impute

- This approach follows the following processes:
  1. For all rows with Kidney  $> 2.5$ ,
  2. For all rows with Liver  $> 60$ ,
  3. Used a K-Nearest Neighbor Imputer to fill in the missing values.
- Leads to less of a significant decrease in error, but still not as good as keeping the values. This means that the outliers in the Kidney and Liver columns are important.
- Below: left table is no outlier removal; right table is with outlier removal.

	Model Name	RMSE	MAPE	R2		Model Name	RMSE	MAPE	R2
0	xgboost	53.677474	0.641035	0.231580	0	xgboost	58.237582	0.641266	0.095473
1	stepwise	58.254232	0.773154	0.094956	1	stepwise	59.255465	0.758722	0.063578
2	jackknife	58.347698	0.784306	0.092049	2	ridge	59.769229	0.758428	0.047269
3	linear	58.347729	0.784303	0.092048	3	elasticnet	59.869616	0.763744	0.044066
4	ridge	59.436676	0.794433	0.057842	4	lasso	60.603372	0.774127	0.020491
5	lasso	59.756936	0.795793	0.047661	5	bayesian	61.085111	0.788345	0.004857
6	elasticnet	60.775126	0.809960	0.014931	6	linear	63.021501	0.734878	-0.059235
7	bayesian	61.583675	0.818436	-0.011454	7	jackknife	63.024816	0.734882	-0.059346
8	polynomial	79.204878	0.879420	-0.673087	8	polynomial	139.271998	1.371672	-4.172999

Figure 4.12: Performance of Mark Outlier Values Outlier Removal Method

### Exacerbate Outliers by Duplicating

- Given that these outlier data points are important, it seems reasonable to make them more pronounced in the dataset.
- By simply duplicating rows that count as outliers for Kidney and Liver features, performance across models improves quite significantly.
- Below: left table is with no outlier removal; right table is with outlier removal.

	<b>Model Name</b>	<b>RMSE</b>	<b>MAPE</b>	<b>R2</b>		<b>Model Name</b>	<b>RMSE</b>	<b>MAPE</b>	<b>R2</b>
<b>0</b>	xgboost	53.677474	0.641035	0.231580	<b>0</b>	xgboost	55.331044	0.682485	0.183507
<b>1</b>	stepwise	58.254232	0.773154	0.094956	<b>1</b>	stepwise	57.929536	0.767494	0.105017
<b>2</b>	jackknife	58.347698	0.784306	0.092049	<b>2</b>	lasso	58.031411	0.779323	0.101866
<b>3</b>	linear	58.347729	0.784303	0.092048	<b>3</b>	linear	58.035470	0.780173	0.101741
<b>4</b>	ridge	59.436676	0.794433	0.057842	<b>4</b>	jackknife	58.035536	0.780181	0.101738
<b>5</b>	lasso	59.756936	0.795793	0.047661	<b>5</b>	elasticnet	58.035802	0.780173	0.101730
<b>6</b>	elasticnet	60.775126	0.809960	0.014931	<b>6</b>	ridge	58.067036	0.780290	0.100763
<b>7</b>	bayesian	61.583675	0.818436	-0.011454	<b>7</b>	bayesian	61.589892	0.817652	-0.011658
<b>8</b>	polynomial	79.204878	0.879420	-0.673087	<b>8</b>	polynomial	71.054859	0.874056	-0.346488

Figure 4.13: Performance of Exacerbate Outliers Removal Method

#### 4.2.7 Qualitative Analysis Techniques

Qualitative data are commonly text-based. However, quantitative data (in the new generated dataset) may be transformed from qualitative data (for example, transforming numerical lab test outputs to two categories: normal and abnormal) and can be analyzed graphically. There are three approaches in qualitative: 1. Inductive (developing a hypothesis or exploratory), 2. Deductive (guided by a hypothesis or explanatory), 3. Combination of inductive and deductive (refining a hypothesis or can be exploratory or explanatory). In this research, we focus on visual approaches and exploratory data analysis.

Figure 4.14: Frequency of top 10 terms in the first year of treatment

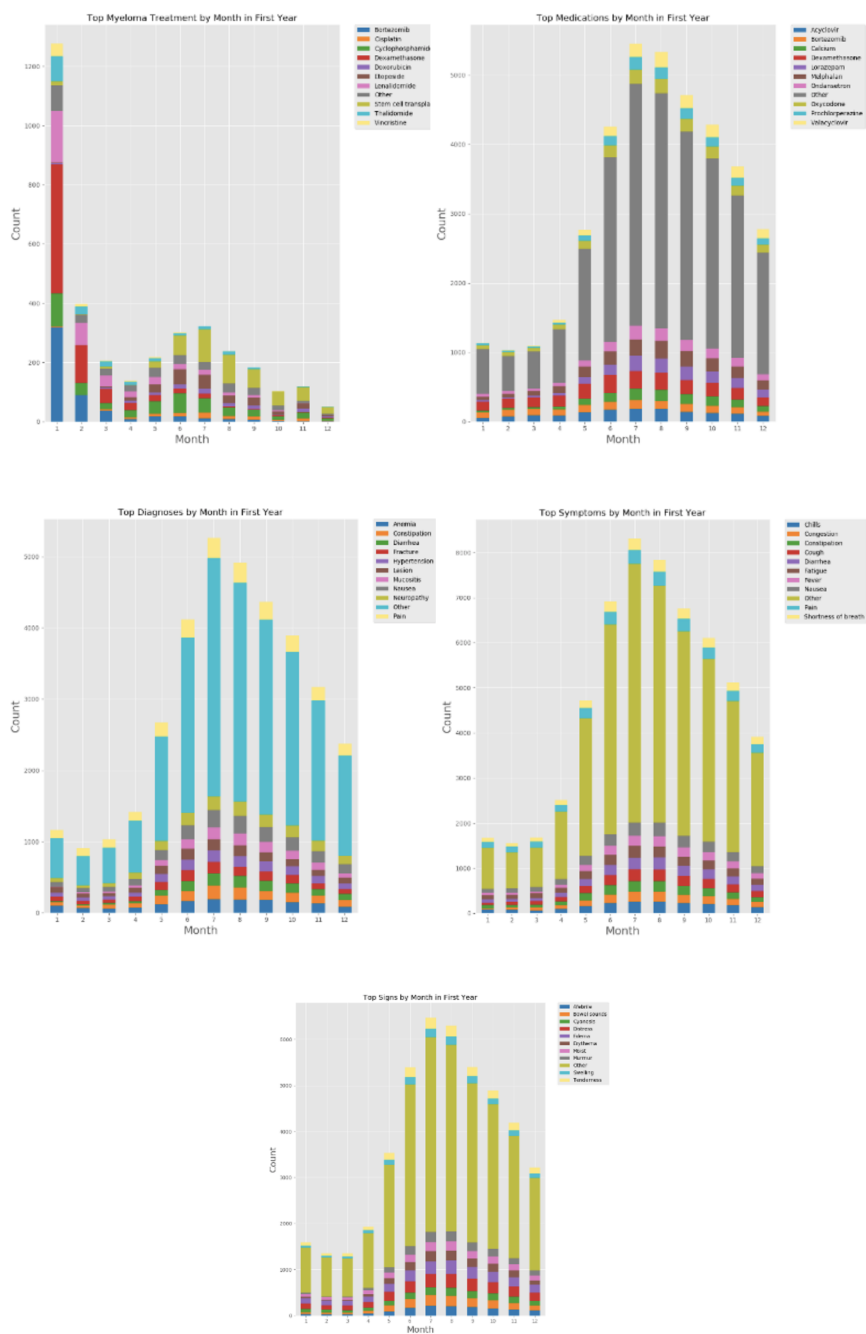


Figure 4.14 depicts the frequency of the top 10 terms in each month for the first year of treatment course in myeloma therapy, medications, diagnoses, symptoms, and signs datasets, respectively.

### **4.3 Result**

#### *4.3.1 Accuracy of Extraction through Amazon Comprehend Medical Service*

In the era of personalized medicine, NLP helps accelerate converting unstructured clinical data into structured data. This technology can be used for oncological-based research where oncologists are interested in implementing NLP for their clinical NLP research. Also, this technology is used for quality improvement.

This section discusses the accuracy of ACM extraction by applying it to our multiple myeloma cancer patients at Seattle Cancer Care Alliance to see if this human-free extraction is better than manual extraction.

Of the automatically extracted records, patients were found to have had an average of 8 diagnoses, 2 billing codes, 10 signs, 8 symptoms, 1 line of treatment, and 8 medications recorded per day of follow up. Before processing, the datasets we focused on contained 376 (billingcodes), 579 (diagnoses), 14 (myelomatherapy), and 398 (medications) unique features that occurred over 100 times in the dataset. After processing, the first pair shared 29 unique features, while the second shared 12. Table 4.4 contains the accuracy of several features from the automated dataset when compared to the manual data. The rest of the data characteristics can be found in Table 4.5.

Both sets of data had many features that were not shared between the automatically and manually compiled datasets. This is most apparent in the diagnoses/billing codes pair, which had a combined 897 unique features that were only present in one of the datasets.

Table 4.4: Accuracy of an assortment of features from the automatically extracted data when compared to the manual dataset

Billing Codes/Diagnoses		Myeloma Therapy/Medications	
Features	% Accuracy	Features	% Accuracy
Acidosis	99.6%	Adriamycin	99.8%
Cardiomegaly	99.7%	Cisplatin	99.7%
Diplopia	99.8%	Elotuzumab	99.4%
End Stage Renal Disease	99.9%	Lenalidomide	99.9%
Hematemesis	99.8%	Prednisone	99.4%

Table 4.5: Characteristics of the manually and automatically constructed datasets

	Diagnoses	Billing Codes	Medications	Myeloma Therapy
# unique features before any processing	4364	5834	1328	67
# unique features in common	281	281	36	36
# unique features with > 100 count	579	376	398	14
# unique features in common with > 100 count	29	29	12	12

ACM does not have any features to detect adverse drug events; therefore, we focused on medication extraction. Table 4.5 compares the manual text extraction for treatments (MyelomaTherapy) and ACM extraction for medications (Medications). To do this, first, we find all unique MedTx in the MyelomaTherapy, then we take the maximum “DayFromDxStart” (the start date of the final phase of therapy). Next, we remove all observations in “DaysFromDx” (the day from diagnosis that a specific medication is taken) in the Medications file that are greater than the maximum “DaysFromDxStart” in the MyelomaTherapy. Also, we eliminate all terms in the Medications file that have null NegationScore and TermScore less than 0.9. The following table shows how many patients take a specific MedTx in each file. Among those patients, this table displays how many and what percentage of them are only in the MyelomaTherapy file, only in the Medications file, and in both

files (see Figure 4.15).

MedTx	MT Mismatch	MT Percentage	MT Total Patients	M Mismatch	M Total Patients	M Percentage	Intersection
<b>Bortezomib</b>	677	96.99	698	0	21	0	21
<b>Cyclophosphamide</b>	586	94.52	620	2	36	5.56	34
<b>Dexamethasone</b>	526	63.83	824	0	298	0	298
<b>Lenalidomide</b>	637	100	637	0	0		0

Figure 4.15: Extraction accuracy in myeloma therapy (MT) and medication (M) datasets

The above table shows the number of mismatched patient IDs and their corresponding percentage between myeloma therapy (manual data extraction) and medication (NLP-ACM data extraction). MT mismatch and M mismatch indicate the number of patients ID that exist in only myeloma therapy and medication, respectively. The MT Percentage and M Percentage columns represent the corresponding percentage. The MT Total Patients and M Total patients columns represent the total number of patients diagnosed by a specific therapy. The intersection column represents the number of common patient IDs in both datasets.

In this study, we find that, while Amazon Comprehend Medical (ACM) provides an excellent way to extract features from unstructured clinical data, it lacks the nuance required for the data to be useful for studies such as retrospective analyses. The findings suggest that, to have a comprehensive model that can fully automate extraction for a potential study, one needs to create the model themselves and familiarize it with the nuances specific to their research question using specific rules. The main outcome is accuracy ( $\sim 68\%$ ). Machine learning algorithms are tested on the test dataset and their performances are evaluated through RMSE, MAPE, and R-Squared error measures.

From our analysis of the dataset extracted using ACM, we found that it would be use-

ful to extract features and their locations in clinical notes. However, the data it extracts is not yet ready for use in retrospective studies that attempt to use the wide range of features to predict clinical outcomes, such as infection with *clostridium difficile* after stem cell transplant and time-to-next-treatment after relapse phase. This is due to two major problems: ACM's failure to introduce sufficient context for the features extracted and the general inconsistencies present in notes written by different providers.

#### **4.4 Discussion**

As research volume steadily increases every year, data collection methods are becoming a topic of increasing interest since assembling a dataset can often be the most cumbersome part of retrospective clinical study. The area with the most promise within automated data collection is NLP, as data science tools to extract information from text are constantly evolving and becoming more powerful. Our research explores one such tool, focused on generalized algorithmic extraction of raw features with NER. Our results show that this application of NER is extremely useful when a set of predefined parameters are applied to the model, including specific feature names and criteria. This is demonstrated by the very high agreement of the automatically extracted features with the manually created dataset.

Although we demonstrated high agreement between automatically and manually extracted data when the feature names were specified, we also showed that when there is not a rigid specification, the datasets vary wildly. Many factors can play into this discrepancy, including erroneously tagged features or a wider set of features captured by the algorithm, or a manual dataset tailored to a specific question which excludes irrelevant data. It should also be noted that we used very strict rules for negativity scores and duplicates. We may have excluded valid data that simply had a low negativity score or missed the context of data that may have been mentioned and recorded multiple times in one physician's note. The algorithm may also have collected data on features that were simply mentioned as a part of a physician's differential but did not make it into the final diagnosis. These elements

would not be missed by a manual reviewer of patient records. With such large datasets, it is impossible to know the exact causes of variance, but it is important to note that it exists and can create potential for bias or incorrect data.

State-of-the-art clinical NER models seem to be very appropriate for retrospective studies where the question is well-defined, with known features of interest. However, the discrepancy between features captured by the model and those collected manually shows that the automated data is not yet ready for exploratory studies where researchers may desire to use all available features in order to identify the relevant variables that influence a clinical outcome.

#### *4.4.1 Feature Context*

The first hurdle faced by researchers wishing to use the data collected by ACM is the fact that ACM, just like many NER models, is designed to simply extract features at face value as they appear in the text, without accounting for most contextual elements that provide additional meaning to the features (such as identifying tests, treatments, and procedures). Failing to understand the context of a feature is one significant drawback of many NER models when compared to human intervention, and it can lead to inconsistencies in the data that may invalidate results. Inconsistent data falls into one of two categories: data that can be easily cleaned and corrected after collection, and data that would require manual revision to verify its accuracy. With the millions of data points in our dataset, any potential issue falling into the second category would introduce a significant need for manual revision, which largely defeats the purpose of using NER to replace manual data collection for retrospective studies.

Examples that fall into the first category include multiple mentions of the same medication on the same note and during the same day even when the medication was only administered

once during that visit. This is actually a feature of ACM instead of a flaw since it is designed to extract every named entity without special rules regarding repetition, and these can easily be reduced to a single data point per day through some simple data cleaning. Another example would be irrelevant terms or phrases that are mistakenly extracted as valid data; phrases such as “worry” extracted as a symptom, or “well nourished” as a diagnosis are clearly noise for most clinical studies and can be easily removed.

Other, much more consequential errors in automated data collections fall into the second category mentioned above and may severely impact the validity of any study conducted using the dataset. For example, when features in the medications dataset (automatically extracted) were compared to features in the myeloma therapy dataset (manually extracted), there were many inconsistencies found between the two, which indicates that the automatically extracted dataset missed a large amount of valuable information and even potentially included false data. Besides missing some data points that should have lined up with the events in the myeloma therapy data, the medications dataset also included extra data points of medications on days that were not recorded manually. The most likely explanation for this is that ACM did not distinguish between cases where a physician mentioned a medication but did not ultimately administer it, and cases where the physician did administer the medication. Another scenario could be one where the physician screened for a disease due to the patient’s symptoms but diagnosed it differently after examining the patient. An NER model would still extract the disease that was initially screened for as if it were present since it was mentioned in the text. These types of contextual extrapolations are a difficult challenge in the world of NER and will require further development to improve the reliability of data extraction.

There have been many studies and endeavors to create comprehensive NLP models to extract relevant clinical information from EMRs. The simple task of extracting terms that are

relevant to a clinical inquiry has been thoroughly addressed and many models, including ACM, can do this quite well by now. However, the challenge to create a model that can comprehensively evaluate the context of features in the text to identify their relevance has only been addressed in very specific scenarios, such as ACM identifying negation and the location of features in the text or other models identifying interactions between medications/drugs. No model has collectively addressed all the nuances in clinical records, which proves to be a major hurdle that must be overcome to truly claim to be an independent and generalized model for clinical data extraction. Even ACM's ability to identify negation is not always reliable because it is difficult to distinguish between family history or the patient when only the entities are provided.

#### *4.4.2 Inconsistencies in Clinical Notes*

The challenge of automating data collection, and data collection in general, is made even more difficult due to the unstructured nature of clinical records, particularly because each provider has their own subtle differences in notetaking. For example, physicians may vary in the extent to which they evaluate a patient during each visit. If a patient is seen for an acute complaint, a physician may not see the need for a full checkup or exam, and this information would be missing from that visit's notes. This can lead to inconsistencies across patients' clinical timelines as well as between patients with different providers because certain features may be overly emphasized in the clinical records at otherwise unimportant timepoints purely because providers mentioned the features more frequently in their notes. It presents the challenge of normalizing the frequency of features to reduce the impact of these inconsistencies while preserving the truly important fluctuations that do have clinical significance. Another inconsistency could occur when some physicians choose only to report positive features and not negative ones, which could skew the data for these patients. Finally, different providers may refer to features differently. As an example, the valid synonyms for multiple myeloma include "mm", "multiple myeloma", and "myeloma". It can quickly

become a daunting task to combine synonyms for the thousands of unique features present in a dataset. There may be small fixes for all of these issues such as normalization of data and feature engineering to remove the biases, but the core of this problem is that modern EMRs are not designed with the intention of being used in research. They are much more oriented towards facilitating billing and documentation, so to truly resolve the inconsistencies in collected data, EMRs would need to become more standardized and research oriented.

#### *4.4.3 Challenges of Using NLP in Oncology*

Many believe that, by applying a packaged clinical NLP mechanism, the structured data can be extracted, and therefore, they consider clinical text mining as a solved problem (Percha [2021], Wahl et al. [2022], Zhou et al. [2006], Dalianis [2014]). There are some limitations of the usage of current NLP techniques, specifically in clinical applications. Although some software systems extract the clinical text information automatically, the performance differs between the systems and depends on the trained data. In the previous sections, we showed the performance of the ACM extraction for named entity recognition and relationship extraction and compared it with the manual data extraction. Although usage of neural network models overcame some of the limitations, these models require large data set to have good performance, but clinical data is small and sparse. As we mentioned above, ACM cannot distinguish medication names that relate to groups of medications (e.g., cardiac medications and antibiotics).

### **4.5 Conclusions**

While ACM provides an excellent way to extract features from unstructured clinical data, it lacks the nuance required for the data to be useful for studies such as retrospective analyses. Currently, to have a comprehensive model that can fully automate extraction for a potential study, one needs to create the model themselves and familiarize it with the nuances specific to their research question using specific rules. This limitation highlights the need for a generalized model that can accurately identify useful features in unstructured records and

be applied to more broad use cases. Such a model will significantly reduce the time needed to carry out studies reliant on previous clinical data, for which the data collection phase often takes the most time. Since recognizing entities has been thoroughly researched, future work should shift to focus on further exploration of the context surrounding the features to extract more meaning from the data.

## Chapter 5

### CONCLUSIONS

In this chapter, we provide a summary of our studies in this thesis. We also suggest some beneficial extensions of our works.

In Chapter 2, we designed a dynamic pricing mechanism to reduce an imbalance problem in a trailer-sharing system subject to a budget limitation using a regret minimization approach. Our proposed incentive-pricing mechanism engages drivers in the facility balancing process. We collaborated with the digital freight company, Convoy, and built pricing and facility-selection algorithms for their platform. We show (by simulation) that our pricing mechanism is empirically efficient compared to the optimal offline market (benchmark) price.

While we focus on trailer-sharing in the trucking industry, similar problems arise in the other domains, such as car/bike/scooter sharing. In car-sharing (e.g., car2go, ReachNow, Zipcar) and bike-sharing (e.g., CitiBike), the similar capability is offered that customers select where to drop-off vehicles/bikes besides the location where they picked up the vehicle/bikes.

We started with a mechanism for the dynamic pricing in Chapter 2 and then extended it to our auction pricing mechanism in Chapter 3. We proved that our bidding model is empirically efficient compared to the dynamic pricing mechanism and dramatically outperforms it. We also proved the truthfulness and individual rationality of our auction pricing mechanism. Our experiments support the practical applicability of our mechanisms on any crowdsourcing platform to increase its efficiency.

It would be useful to extend our approach to design mechanisms which can take into account different utility functions for both Convoy and the drivers on the platform. It would be interesting to show their theoretical guarantees. In this work, we assumed a large pool of drivers. Perhaps a more natural approach is to assume a limited number of drivers, as well as a limited amount of time.

In Chapter 4 of this thesis, we focused on the application of machine learning and natural language processing in the healthcare domain for multiple myeloma cancer patients in Fred Hutchinson Cancer Research Center. In this chapter, we evaluated whether ACM extracts entities from clinical notes and identifies medical information better than a human. We found that, while ACM provides an excellent way to extract features from unstructured clinical data, it lacks the nuance required for the data to be useful for studies such as retrospective analyses. The findings suggest that, to have a comprehensive model that can fully automate extraction for a potential study, one needs to create the model themselves and familiarize it with the nuances specific to their research question using specific rules. This limitation highlights the need for a generalized model that can accurately identify useful features in unstructured records and be applied to more broad use cases. Such a model will significantly reduce the time needed to carry out studies reliant on previous clinical data, for which the data collection phase often takes the most time. Since recognizing entities has been thoroughly researched, future work should shift focus onto further exploring context of the text surrounding features to extract more meaning from the data.

## Bibliography

- Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. Advances in neural information processing systems, 24, 2011.
- I. Abraham, O. Alonso, V. Kandylas, and A. Slivkins. Adaptive crowdsourcing algorithms for the bandit survey problem. In Conference on learning theory, pages 882–910. PMLR, 2013.
- G. Aggarwal, G. Goel, C. Karande, and A. Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms, pages 1253–1264. SIAM, 2011.
- O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafir. The rise of raas: the resource-as-a-service cloud. Communications of the ACM, 57(7):76–84, 2014.
- S. Agrawal and N. Goyal. Further optimal regret bounds for thompson sampling. In Artificial intelligence and statistics, pages 99–107. PMLR, 2013.
- J. Alonso-Mora, A. Wallar, and D. Rus. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3583–3590. IEEE, 2017.
- W. Amaldoss and S. Jain. Joint bidding in the name-your-own-price channel: A strategic analysis. Management Science, 54(10):1685–1699, 2008.

- K. Amin, M. Kearns, P. Key, and A. Schwaighofer. Budget optimization for sponsored search: Censored learning in mdps. arXiv preprint arXiv:1210.4847, 2012a.
- K. Amin, M. Kearns, P. Key, and A. Schwaighofer. Budget optimization for sponsored search: Censored learning in mdps. arXiv preprint arXiv:1210.4847, 2012b.
- P. M. Anglin, R. Rutherford, and T. M. Springer. The trade-off between the selling price of residential properties and time-on-the-market: The impact of price setting. The Journal of Real Estate Finance and Economics, 26(1):95–111, 2003.
- D. Ardagna, S. Casolari, and B. Panicucci. Flexible distributed capacity allocation and load redirect algorithms for cloud systems. In 2011 IEEE 4th International Conference on Cloud Computing, pages 163–170. IEEE, 2011.
- J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In COLT, pages 41–53. Citeseer, 2010.
- P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2):235–256, 2002a.
- P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2):235–256, 2002b.
- P. Auer, R. Ortner, and C. Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In International Conference on Computational Learning Theory, pages 454–468. Springer, 2007.
- P. Azar, S. Micali, C. Daskalakis, and S. M. Weinberg. Optimal and efficient parametric auctions. In Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms, pages 596–604. SIAM, 2013.

- M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. A knapsack secretary problem with applications. In Approximation, randomization, and combinatorial optimization. Algorithms and techniques, pages 16–28. Springer, 2007.
- M. Babaioff, M. Dinitz, A. Gupta, N. Immorlica, and K. Talwar. Secretary problems: weights and discounts. In Proceedings of the twentieth annual ACM-SIAM Symposium on Discrete Algorithms, pages 1245–1254. SIAM, 2009.
- M. Babaioff, S. Dughmi, R. D. Kleinberg, and A. Slivkins. Dynamic pricing with limited supply. 3 (1): 4, 2015. Special issue for 13th ACM EC, 2012.
- M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms. SIAM Journal on Computing, 43(1):194–230, 2014.
- A. Badanidiyuru, R. Kleinberg, and Y. Singer. Learning on a budget: posted price mechanisms for online procurement. In Proceedings of the 13th ACM conference on electronic commerce, pages 128–145, 2012.
- S. Banerjee, R. Johari, and C. Riquelme. Pricing in ride-sharing platforms: A queueing-theoretic approach. In Proceedings of the Sixteenth ACM Conference on Economics and Computation, pages 639–639, 2015.
- Z. Bar-Yossef, K. Hildrum, and F. Wu. Incentive-compatible online auctions for digital goods. In SODA, volume 2, pages 964–970, 2002.
- M. Bayati, N. Hamidi, R. Johari, and K. Khosravi. Unreasonable effectiveness of greedy algorithms in multi-armed bandit with many arms. Advances in Neural Information Processing Systems, 33:1713–1723, 2020.
- X. Bei, N. Chen, N. Gravin, and P. Lu. Budget feasible mechanism design: from prior-free

- to bayesian. In Proceedings of the forty-fourth annual ACM symposium on Theory of computing, pages 449–458, 2012.
- E. Belavina, S. Marinesi, and G. Tsoukalas. Rethinking crowdfunding platform design: mechanisms to deter misconduct and improve efficiency. Management Science, 66(11):4980–4997, 2020.
- S. Benjaafar, G. Kong, X. Li, and C. Courcoubetis. Peer-to-peer product sharing: Implications for ownership, usage, and social welfare in the sharing economy. Management Science, 65(2):477–493, 2019.
- M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soyent: a word processor with a crowd inside. In Proceedings of the 23rd annual ACM symposium on User interface software and technology, pages 313–322, 2010.
- D. A. Berry and B. Fristedt. Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). London: Chapman and Hall, 5(71-87):7–7, 1985.
- O. Besbes and A. Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. Operations Research, 57(6):1407–1420, 2009.
- K. Bimpikis, O. Candogan, and D. Saban. Spatial pricing in ride-sharing networks. Operations Research, 67(3):744–769, 2019.
- A. Biswas, S. Jain, D. Mandal, and Y. Narahari. A truthful budget feasible multi-armed bandit mechanism for crowdsourcing time critical tasks. In AAMAS, pages 1101–1109, 2015.
- A. Blum and J. D. Hartline. Near-optimal online auctions. 2005.

- A. Blum and Y. Mansour. Learning, regret minimization, and equilibria. 2007.
- A. Blum, V. Kumar, A. Rudra, and F. Wu. Online learning in online auctions. Theoretical Computer Science, 324(2-3):137–146, 2004.
- T. Bonald and A. Proutiere. Two-target algorithms for infinite-armed bandits with bernoulli rewards. In NIPS 2013-Neural Information Processing Systems Conference, page 8, 2013.
- D. C. Brabham. Crowdsourcing. Mit Press, 2013.
- M. Braverman, J. Mao, J. Schneider, and S. M. Weinberg. Multi-armed bandit problems with strategic arms. In Conference on Learning Theory, pages 383–416. PMLR, 2019.
- P. Brown, H. Lauder, and D. Ashton. The global auction: The broken promises of education, jobs, and incomes. Oxford University Press, 2010.
- S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends in Machine Learning, 5(1):1–122, 2012.
- S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In International conference on Algorithmic learning theory, pages 23–37. Springer, 2009.
- N. Buchbinder, K. Jain, and J. S. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In European Symposium on Algorithms, pages 253–264. Springer, 2007.
- G. P. Cachon. A research framework for business models: what is common among fast fashion, e-tailing, and ride sharing? Management science, 66(3):1172–1192, 2020.
- G. P. Cachon, K. M. Daniels, and R. Lobel. The role of surge pricing on a service platform with self-scheduling capacity. Manufacturing & Service Operations Management, 19(3):368–384, 2017.

- A. Carpentier and M. Valko. Simple regret for infinitely many armed bandits. In International Conference on Machine Learning, pages 1133–1141. PMLR, 2015.
- Carter-Logistics. Self-driving trucks and the future of us supply chains, 2021. URL <https://carter-logistics.com/self-driving-trucks-and-the-future-of-us-supply-chains/>.
- CCJ-Digital. Let excitement, not fear, drive your perceptions of autonomous trucks, 2021. URL <https://www.ccjdigital.com/equipment-controls/article/15042919/trucking-industrys-future-with-autonomous-trucks>.
- E. M. Cepolina and A. Farina. Urban car sharing: An overview of relocation strategies. WIT Transactions on the Built Environment, 128:419–431, 2012.
- T. Chakraborty, E. Even-Dar, S. Guha, Y. Mansour, and S. Muthukrishnan. Selective call out and real time bidding. In International Workshop on Internet and Network Economics, pages 145–157. Springer, 2010.
- T.-S. Chang. Decision support for truckload carriers in one-shot combinatorial auctions. Transportation Research Part B: Methodological, 43(5):522–541, 2009.
- O. Chapelle and L. Li. An empirical evaluation of thompson sampling. Advances in neural information processing systems, 24:2249–2257, 2011.
- D. Chemla, F. Meunier, and R. W. Calvo. Bike sharing systems: Solving the static rebalancing problem. Discrete Optimization, 10(2):120–146, 2013.
- N. Chen, N. Gravin, and P. Lu. On the approximability of budget feasible mechanisms. In Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms, pages 685–699. SIAM, 2011.

- R. L.-Y. Chen, S. AhmadBeygi, A. Cohn, D. R. Beil, and A. Sinha. Solving truckload procurement auctions over an exponential number of bundles. Transportation Science, 43(4):493–510, 2009.
- Y. Chen, B. Li, and Q. Zhang. Incentivizing crowdsourcing systems with network effects. In IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, pages 1–9. IEEE, 2016.
- Z. Chen, K. Liu, J. Wang, and T. Yamamoto. H-convlstm-based bagging learning approach for ride-hailing demand prediction considering imbalance problems and sparse uncertainty. Transportation Research Part C: Emerging Technologies, 140:103709, 2022.
- M. Clemente, M. P. Fanti, A. M. Mangini, and W. Ukovich. The vehicle relocation problem in car sharing systems: Modeling and simulation in a petri net framework. In International Conference on Applications and Theory of Petri Nets and Concurrency, pages 250–269. Springer, 2013.
- Convoy. Transport the world with endless capacity and zero waste, 2022. URL <https://convoy.com/about-us/>.
- Convoy-Team. Convoy launches dedicated freight for owner-operators and small fleets nationwide, 2020. URL <https://convoy.com/blog/convoy-launches-dedicated-freight/>.
- L. DaCosta, A. Fialho, M. Schoenauer, and M. Sebag. Adaptive operator selection with dynamic multi-armed bandits. In Proceedings of the 10th annual conference on Genetic and evolutionary computation, pages 913–920, 2008.
- H. Dalianis. Clinical text retrieval-an overview of basic building blocks and applications. Professional Search in the Modern World, pages 147–165, 2014.

- Y. David and N. Shimkin. Infinitely many-armed bandits with unknown value distribution. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 307–322. Springer, 2014.
- G. H. de Almeida Correia and A. P. Antunes. Optimization approach to depot location and trip selection in one-way carsharing systems. Transportation Research Part E: Logistics and Transportation Review, 48(1):233–247, 2012.
- S. De Vries and R. V. Vohra. Combinatorial auctions: A survey. INFORMS Journal on computing, 15(3):284–309, 2003.
- Deloitte-Insights. Autonomous trucks lead the way, 2021. URL <https://www2.deloitte.com/us/en/insights/focus/future-of-mobility/\linebreakautonomous-trucks-lead-the-way.html>.
- N. R. Devanur and J. D. Hartline. Limited and online supply and the bayesian foundations of prior-free mechanism design. In Proceedings of the 10th ACM conference on Electronic commerce, pages 41–50, 2009.
- N. R. Devanur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In Proceedings of the 10th ACM conference on Electronic commerce, pages 71–78, 2009a.
- N. R. Devanur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In Proceedings of the 10th ACM conference on Electronic commerce, pages 71–78, 2009b.
- N. R. Devanur and S. M. Kakade. The price of truthfulness for pay-per-click auctions. In Proceedings of the 10th ACM conference on Electronic commerce, pages 99–106, 2009.

- N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In Proceedings of the 12th ACM conference on Electronic commerce, pages 29–38, 2011.
- N. R. Devanur, K. Jain, and R. D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms, pages 101–107. SIAM, 2013.
- W. Ding, T. Qin, X.-D. Zhang, and T.-Y. Liu. Multi-armed bandit with budget constraint and variable costs. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 27, 2013.
- A. Drutsa. Reserve pricing in repeated second-price auctions with strategic bidders. In International Conference on Machine Learning, pages 2678–2689. PMLR, 2020.
- M. L. Emiliani and D. J. Stec. Realizing savings from online reverse auctions. Supply Chain Management: An International Journal, 2002.
- L. Esterle. Goal-aware team affiliation in collectives of autonomous robots. In 2018 IEEE 12th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), pages 90–99. IEEE, 2018.
- H. Etzion, E. Pinker, and A. Seidmann. Analyzing the simultaneous use of auctions and posted prices for online selling. Manufacturing & Service Operations Management, 8(1): 68–91, 2006.
- Y. Fan, H. Sun, and X. Liu. Truthful incentive mechanisms for dynamic and heterogeneous tasks in mobile crowdsourcing. In 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), pages 881–888. IEEE, 2015.

- R. Faullant, J. Fueller, and K. Hutter. Fair play: perceived fairness in crowdsourcing communities and its behavioral consequences. In Academy of Management Proceedings, volume 2013, page 15433. Academy of Management Briarcliff Manor, NY 10510, 2013.
- S. Fay. Partial-repeat-bidding in the name-your-own-price channel. Marketing Science, 23(3):407–418, 2004.
- J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein. Online stochastic packing applied to display ad allocation. In European Symposium on Algorithms, pages 182–194. Springer, 2010.
- M. A. Figliozzi, H. S. Mahmassani, and P. Jaillet. Pricing in dynamic vehicle routing problems. Transportation Science, 41(3):302–318, 2007.
- S. Filippi, O. Cappé, and A. Garivier. Optimism in reinforcement learning and kullback-leibler divergence. In 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pages 115–122. IEEE, 2010.
- A. Flajolet and P. Jaillet. Real-time bidding with side information. 2017.
- N. Franke, P. Keinz, and K. Klausberger. “does this sound like a fair deal?”: Antecedents and consequences of fairness expectations in the individual’s decision to participate in firm innovation. Organization science, 24(5):1495–1516, 2013.
- C. Fricker, N. Gast, and H. Mohamed. Mean field analysis for inhomogeneous bike sharing systems. In Discrete Mathematics and Theoretical Computer Science, 2012.
- A. Garivier and O. Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In Proceedings of the 24th annual conference on learning theory, pages 359–376. JMLR Workshop and Conference Proceedings, 2011.

- D. K. George and C. H. Xia. Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. European journal of operational research, 211(1):198–207, 2011.
- J. Gittins. Multi-armed bandit allocation indices wiley-interscience series in systems and optimization (chichester: John wiley & sons, ltd.). 1989.
- J. Gittins, K. Glazebrook, and R. Weber. Multi-armed bandit allocation indices. John Wiley & Sons, 2011.
- A. Gneezy, A. Imas, A. Brown, L. D. Nelson, and M. I. Norton. Paying to be nice: Consistency and costly prosocial behavior. Management Science, 58(1):179–187, 2012.
- G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In SODA, volume 8, pages 982–991, 2008.
- R. Gonen and E. Pavlov. An incentive-compatible multi-armed bandit mechanism. In Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing, pages 362–363, 2007.
- M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes. Adaptive limited-supply online auctions. In Proceedings of the 5th ACM Conference on Electronic Commerce, pages 71–80, 2004.
- M. Haliem, G. Mani, V. Aggarwal, and B. Bhargava. A distributed model-free ride-sharing algorithm with pricing using deep reinforcement learning. In Computer Science in Cars Symposium, pages 1–10, 2020.
- I.-H. Hann and C. Terwiesch. Measuring the frictional costs of online transactions: The case of a name-your-own-price channel. Management Science, 49(11):1563–1579, 2003.

- J. J. Horton and L. B. Chilton. The labor economics of paid crowdsourcing. In Proceedings of the 11th ACM conference on Electronic commerce, pages 209–218, 2010.
- J. J. Horton and R. J. Zeckhauser. Algorithmic wage negotiations: Applications to paid crowdsourcing. Proceedings of CrowdConf, 4:2–5, 2010.
- J. Huang, R. Kong, and H. Zhou. Online learning in channel sensing order for cognitive radios. 2014.
- T. Hubert. Announcing convoy go, a nationwide drop hook marketplace, 2019. URL <https://convoy.com/blog/announcing-convoy-go-a-nationwide-drop-hook-marketplace/>.
- J. Jia, Q. Zhang, Q. Zhang, and M. Liu. Revenue generation for truthful spectrum auction in dynamic spectrum access. In Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing, pages 3–12, 2009.
- Z. Karnin, T. Koren, and O. Somekh. Almost optimal exploration in multi-armed bandits. In International Conference on Machine Learning, pages 1238–1246. PMLR, 2013.
- M. L. Kasavana and A. Singh. Online auctions: dynamic pricing and the lodging industry. Journal of Hospitality & Leisure Marketing, 9(3-4):127–140, 2001.
- E. Kaufmann, O. Cappé, and A. Garivier. On bayesian upper confidence bounds for bandit problems. In Artificial intelligence and statistics, pages 592–600. PMLR, 2012a.
- E. Kaufmann, N. Korda, and R. Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In International conference on algorithmic learning theory, pages 199–213. Springer, 2012b.

- A. G. Kek, R. L. Cheu, Q. Meng, and C. H. Fung. A decision support system for vehicle relocation operations in carsharing systems. Transportation Research Part E: Logistics and Transportation Review, 45(1):149–158, 2009.
- M. R. Khadge and M. V. Kulkarni. Machine learning approach for predicting end price of online auction. In 2016 International Conference on Inventive Computation Technologies (ICICT), volume 3, pages 1–5. IEEE, 2016.
- R. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. Advances in Neural Information Processing Systems, 17:697–704, 2004a.
- R. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. Advances in Neural Information Processing Systems, 17, 2004b.
- R. Kleinberg and T. Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In 44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings., pages 594–605. IEEE, 2003.
- D. Koolen, N. Sadat-Razavi, and W. Ketter. Machine learning for identifying demand patterns of home energy management systems with dynamic electricity pricing. Applied Sciences, 7(11):1160, 2017.
- R. Kumar, B. Moseley, S. Vassilvitskii, and A. Vattani. Fast greedy algorithms in mapreduce and streaming. ACM Transactions on Parallel Computing (TOPC), 2(3):1–22, 2015.
- A. Lacerda, R. L. Santos, A. Veloso, and N. Ziviani. Improving daily deals recommendation using explore-then-exploit strategies. Information Retrieval Journal, 18(2):95–122, 2015.
- T. L. Lai, H. Robbins, et al. Asymptotically efficient adaptive allocation rules. Advances in applied mathematics, 6(1):4–22, 1985.

- T. Lattimore and C. Szepesvári. Bandit algorithms. Cambridge University Press, 2020.
- R. Lavi and N. Nisan. Competitive analysis of incentive compatible on-line auctions. Theoretical Computer Science, 310(1-3):159–180, 2004.
- R. Lavi and N. Nisan. Online ascending auctions for gradually expiring items. 2005.
- C.-G. Lee, R. H. Kwon, and Z. Ma. A carrier’s optimal bid generation problem in combinatorial auctions for transportation procurement. Transportation Research Part E: Logistics and Transportation Review, 43(2):173–191, 2007.
- R. Li, C. Peng, and H. Sun. Product selection strategy analysis of crowdsourcing platform from the full cost perspective. In Proceedings of the 2019 International Conference on Management Science and Industrial Engineering, pages 92–97, 2019.
- C. H. Lin, M. Mausam, and D. S. Weld. Dynamically switching between synergistic workflows for crowdsourcing. In Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.
- C. Lindsey, A. Frei, H. Mahmassani, H. Alibabai, Y. Park, D. Klabjan, M. Reed, G. Langheim, and T. Keating. Online pricing and capacity sourcing for third-party logistics providers. Northwestern Univ., Evanston, IL, USA, Tech. Rep, 2013.
- G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Turkit: tools for iterative tasks on mechanical turk. In Proceedings of the ACM SIGKDD workshop on human computation, pages 29–30, 2009.
- Y.-E. Liu, T. Mandel, E. Brunskill, and Z. Popovic. Trading off scientific knowledge and user learning with multi-armed bandits. In EDM, pages 161–168, 2014.

- L. Luncean, C. Bădică, and A. Bădică. Agent-based system for brokering of logistics services—initial report. In Asian Conference on Intelligent Information and Database Systems, pages 485–494. Springer, 2014.
- C. Maddox. The importance of pricing for profit. Due.com, 2017.
- R. Maestre, J. Duque, A. Rubio, and J. Arévalo. Reinforcement learning for fair dynamic pricing. In Proceedings of SAI Intelligent Systems Conference, pages 120–135. Springer, 2018.
- S. Mannor and J. N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. Journal of Machine Learning Research, 5(Jun):623–648, 2004.
- V. Manshadi and S. Rodilitz. Online policies for efficient volunteer crowdsourcing. Management Science, 2022.
- W. Mason and D. J. Watts. Financial incentives and the” performance of crowds”. In Proceedings of the ACM SIGKDD workshop on human computation, pages 77–85, 2009.
- N. Mehta, P. Detroja, and A. Agashe. Amazon changes prices on its products about every 10 minutes — here’s how and why they do it. INSIDER, 2018.
- S. Miao and X. Chao. Dynamic joint assortment and pricing optimization with demand learning. Manufacturing & Service Operations Management, 23(2):525–545, 2021.
- R. Mohammed. Use pricing strategy to boost sales. Harvard Business Review, 2012.
- K. B. Monroe. Buyers’ subjective perceptions of price. Journal of marketing research, 10(1):70–80, 1973.
- T. Mukhopadhyay and S. Kekre. Strategic and operational benefits of electronic integration in b2b procurement processes. Management science, 48(10):1301–1313, 2002.

- R. B. Myerson. Optimal auction design. Mathematics of operations research, 6(1):58–73, 1981.
- R. Nair and E. Miller-Hooks. Fleet management for vehicle sharing operations. Transportation Science, 45(4):524–540, 2011.
- H. Nazerzadeh, A. Saberi, and R. Vohra. Dynamic cpa mechanisms and applications to online advertising. In ACM Conf. on the World Wide Web, volume 10, 2008.
- M. M. Nejad, L. Mashayekhy, and D. Grosu. Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds. IEEE transactions on parallel and distributed systems, 26(2):594–603, 2014.
- U. D. of Commerce. U.s. census bureau news. 2022.
- Ö. Özer and R. Phillips. The Oxford handbook of pricing management. OUP Oxford, 2012.
- C. Papadimitriou and Y. Singer. Budget feasible mechanisms. CoRR, abs/1002.2334, 2010.
- Y. Papanastasiou, K. Bimpikis, and N. Savva. Crowdsourcing exploration. Management Science, 64(4):1727–1746, 2018.
- B. Percha. Modern clinical text mining: a guide and review. 2021.
- J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari. Dynamic vehicle redistribution and online price incentives in shared mobility systems. IEEE Transactions on Intelligent Transportation Systems, 15(4):1567–1578, 2014.
- J. Pohlisch. Managing the crowd: a literature review of empirical studies on internal crowdsourcing. Internal Crowdsourcing in Companies, page 27, 2021.

- J. Qin, Y. Chow, J. Yang, and R. Rajagopal. Online modified greedy algorithm for storage control under uncertainty. IEEE Transactions on Power Systems, 31(3):1729–1743, 2015.
- C. Qiu, A. Squicciarini, and B. Hanrahan. Incentivizing distributive fairness for crowdsourcing workers. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pages 404–412, 2019.
- K. Raharjo. Applying Multi-Armed Bandit on Game Development. PhD thesis, PhD thesis, University of British Columbia, Okanagan, 2016.
- S. D. Ramchurn, T. D. Huynh, M. Venanzi, and B. Shi. Collabmap: crowdsourcing maps for emergency planning. In Proceedings of the 5th Annual ACM Web Science Conference, pages 326–335, 2013.
- T. Raviv and O. Kolka. Optimal inventory management of a bike-sharing station. IEEE Transactions, 45(10):1077–1093, 2013.
- H. Robbins. Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society, 58(5):527–535, 1952.
- K. Saito, A. Notsu, S. Ubukata, and K. Honda. Performance investigation of ucb policy in q-learning. In 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pages 777–780. IEEE, 2015.
- J. L. Schenker. The platform economy, year = 2019, url = <https://innovator.news/the-platform-economy-3c09439b56>.
- J. L. Schenker. The platform economy, 2019. URL <https://innovator.news/the-platform-economy-3c09439b56>.

- A. Scheuermann and J. Hoxha. Ontologies for intelligent provision of logistics services. City, 2012.
- J. Schuijbroek, R. C. Hampshire, and W.-J. Van Hoesve. Inventory rebalancing and vehicle routing in bike sharing systems. European Journal of Operational Research, 257(3):992–1004, 2017.
- Schumpeter. Flexible figures: A growing number of companies are using “dynamic” pricing. The Economist, 2016.
- A. D. Shaw, J. J. Horton, and D. L. Chen. Designing incentives for inexpert human raters. In Proceedings of the ACM 2011 conference on Computer supported cooperative work, pages 275–284, 2011.
- W. Shen, B. Peng, H. Liu, M. Zhang, R. Qian, Y. Hong, Z. Guo, Z. Ding, P. Lu, and P. Tang. Reinforcement mechanism design: With applications to dynamic pricing in sponsored search auctions. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 2236–2243, 2020.
- A. Simonetto, J. Monteil, and C. Gambella. Real-time city-scale ridesharing via linear assignment problems. Transportation Research Part C: Emerging Technologies, 101:208–232, 2019.
- U. Simonsohn and D. Ariely. When rational sellers face nonrational buyers: evidence from herding on ebay. Management science, 54(9):1624–1637, 2008.
- Y. Singer. @articlepapadimitriou2010budget, title=Budget feasible mechanisms, author=Papadimitriou, Christos and Singer, Yaron, journal=CoRR, abs/1002.2334, year=2010, publisher=Citeseer s. In 2010 IEEE 51st Annual Symposium on foundations of computer science, pages 765–774. IEEE, 2010.

- Y. Singer. How to win friends and influence people, truthfully: influence maximization mechanisms for social networks. In Proceedings of the fifth ACM international conference on Web search and data mining, pages 733–742, 2012.
- Y. Singer and M. Mittal. Pricing tasks in online labor markets. In Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.
- Y. Singer and M. Mittal. Pricing mechanisms for crowdsourcing markets. In Proceedings of the 22nd international conference on World Wide Web, pages 1157–1166, 2013.
- M. Spann and G. J. Tellis. Does the internet promote better consumer decisions? the case of name-your-own-price auctions. Journal of marketing, 70(1):65–78, 2006.
- N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. ICML, page 1015– 1022, 2010.
- P. Ströhle, C. M. Flath, and J. Gärttner. Leveraging customer flexibility for car-sharing fleet optimization. Transportation Science, 53(1):42–61, 2019.
- R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction, 2011.
- C. S. Tang and S. Netessine. Consumer-driven demand and operations management models: a systematic study of information-technology-enabled sales mechanisms. Springer, 2009.
- L. Tang, R. Rosales, A. Singh, and D. Agarwal. Automatic ad format selection via contextual bandits. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pages 1587–1594, 2013.
- C. Terwiesch, S. Savin, and I.-H. Hann. Online haggling at a name-your-own-price retailer: Theory and application. Management Science, 51(3):339–351, 2005.

- W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika, 25(3-4):285–294, 1933.
- H. To, L. Fan, L. Tran, and C. Shahabi. Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints. In 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom), pages 1–8. IEEE, 2016.
- O. Tokarchuk, R. Cuel, and M. Zamarian. Analyzing crowd labor and designing incentives for humans in the loop. IEEE Internet Computing, 16(5):45–51, 2012.
- L. Tran-Thanh, A. Chapman, E. M. De Cote, A. Rogers, and N. R. Jennings. Epsilon–first policies for budget–limited multi–armed bandits. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 24, 2010.
- L. Tran-Thanh, A. Chapman, A. Rogers, and N. Jennings. Knapsack based optimal policies for budget–limited multi–armed bandits. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 26, 2012a.
- L. Tran-Thanh, A. Chapman, A. Rogers, and N. Jennings. Knapsack based optimal policies for budget–limited multi–armed bandits. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 26, pages 1134–1140, 2012b.
- L. Tran-Thanh, L. Stavrogiannis, V. Naroditskiy, V. Robu, N. R. Jennings, and P. Key. Efficient regret bounds for online bid optimisation in budget-limited sponsored search auctions. 2014.
- F. Trovo, S. Paladino, M. Restelli, and N. Gatti. Improving multi-armed bandit algorithms for pricing. In 16th European Conference on Multi-Agent Systems, pages 1–15, 2018.
- M.-T. Tsai, J.-D. Saphores, and A. Regan. Valuation of freight transportation contracts

- under uncertainty. Transportation Research Part E: Logistics and Transportation Review, 47(6):920–932, 2011.
- H. T. Tse and H.-f. Leung. Exploiting semantic epsilon greedy exploration strategy in multi-agent reinforcement learning. arXiv preprint arXiv:2201.10803, 2022.
- B. Turan, R. Pedarsani, and M. Alizadeh. Dynamic pricing and fleet management for electric autonomous mobility on demand systems. Transportation Research Part C: Emerging Technologies, 121:102829, 2020.
- R. Van Tulder and M. Mol. Reverse auctions or auctions reversed:: First experiments by philips. European Management Journal, 20(5):447–456, 2002.
- H. P. Vanchinathan, A. Marfurt, C.-A. Robelin, D. Kossmann, and A. Krause. Discovering valuable items from massive data. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1195–1204, 2015.
- J. Wahl, J. Füller, and K. Hutter. What’s the problem? how crowdsourcing and text-mining may contribute to the understanding of unprecedented problems such as covid-19. R&D Management, 52(2):427–446, 2022.
- R. Wang. Auctions versus posted-price selling. The American Economic Review, pages 838–851, 1993.
- T. Wang, E. Gal-Or, and R. Chatterjee. The name-your-own-price channel in the travel industry: An analytical exploration. Management Science, 55(6):968–979, 2009.
- X. Wang and M. Xia. Combinatorial bid generation problem for transportation service procurement. Transportation research record, 1923(1):189–198, 2005.

- Y. Wei, Y. Zhu, H. Zhu, Q. Zhang, and G. Xue. Truthful online double auctions for dynamic mobile crowdsourcing. In 2015 IEEE Conference on Computer Communications (INFOCOM), pages 2074–2082. IEEE, 2015.
- Y. Wu, F. Li, L. Ma, Y. Xie, T. Li, and Y. Wang. A context-aware multiarmed bandit incentive mechanism for mobile crowd sensing systems. IEEE Internet of Things Journal, 6(5):7648–7658, 2019.
- W. Xia and R. Yap. Learning robust search strategies using a bandit-based approach. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.
- Y. Xia, H. Li, T. Qin, N. Yu, and T.-Y. Liu. Thompson sampling for budgeted multi-armed bandits. arXiv preprint arXiv:1505.00146, 2015.
- Y. Xia, W. Ding, X.-D. Zhang, N. Yu, and T. Qin. Budgeted bandit problems with continuous random costs. In Asian conference on machine learning, pages 317–332. PMLR, 2016.
- M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang. Online task assignment for crowdsensing in predictable mobile social networks. IEEE Transactions on Mobile Computing, 16(8):2306–2320, 2016.
- M. Xiu-li, W. Yi-fan, and L. Bo. Crowdsourcing logistics service quality network equilibrium model with penalty policy. Operations Research and Management Science, 31(2):231, 2022.
- D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In Proceedings of the 18th annual international conference on Mobile computing and networking, pages 173–184, 2012.

- H. Zhang, H. Jiang, B. Li, F. Liu, A. V. Vasilakos, and J. Liu. A framework for truthful online auctions in cloud computing with heterogeneous user demands. IEEE Transactions on Computers, 65(3):805–818, 2015.
- J. Zhang, N. Xie, X. Zhang, K. Yue, W. Li, and D. Kumar. Machine learning based resource allocation of cloud computing in auction. Comput. Mater. Continua, 56(1):123–135, 2018.
- L. Zhang, Z. Li, and C. Wu. Dynamic resource provisioning in cloud computing: A randomized auction approach. In IEEE INFOCOM 2014-IEEE Conference on Computer Communications, pages 433–441. IEEE, 2014.
- D. Zhao, H. Ma, and L. Liu. Frugal online incentive mechanisms for mobile crowd sensing. IEEE Transactions on Vehicular Technology, 66(4):3319–3330, 2016.
- X. Zhou, H. Han, I. Chankai, A. Prestrud, and A. Brooks. Approaches to text mining for clinical medical records. In Proceedings of the 2006 ACM symposium on Applied computing, pages 235–239, 2006.
- Z. Zhu, L. Huang, and H. Xu. Contextual combinatorial cascading thompson sampling. In International Conference on Wireless Algorithms, Systems, and Applications, pages 520–532. Springer, 2019.
- J. Zimmert and Y. Seldin. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. J. Mach. Learn. Res., 22(28):1–49, 2021.