Reinforcement Mechanism Design*

Pingzhong Tang
IIIS, Tsinghua University, China
kenshin@tsinghua.edu.cn

Abstract

We put forward a modeling and algorithmic framework to design and optimize mechanisms in dynamic industrial environments where a designer can make use of the data generated in the process to automatically improve future design. Our solution, coined *reinforcement mechanism design*, is rooted in game theory but incorporates recent AI techniques to get rid of nonrealistic modeling assumptions and to make automated optimization feasible. We instantiate our framework on the key application scenarios of Baidu and Taobao, two of the largest mobile app companies in China. For the Taobao case, our framework automatically designs mechanisms that allocate buyer impressions for the e-commerce website; for the Baidu case, our framework automatically designs dynamic reserve pricing schemes of advertisement auctions of the search engine. Experiments show that our solutions outperform the state-of-the-art alternatives and those currently deployed, under both scenarios.

1 Introduction

Over the past decade, China has become one of the leading countries of smart-phone usage: as of 2016, the total number of daily active users on Wechat, the largest instant messaging application in China, is around 800 million and more than half of this amount spend at least 1.5 hours daily on this app alone; the total gross transaction volume of Taobao (Nasdaq: BABA), the largest e-commerce platform in China, is around 426 billion US dollars. To put the number in perspective, this is more than twice of the total transaction volume of eBay and Amazon combined in the same year. Massive user impressions brought by gigantic scales present huge business opportunities to these app companies.

To monetize these impressions, all these companies have adopted some sort of economic mechanisms (to be described in detail shortly) that allocates these impressions to interested parties (advertisers for search engines and retail sellers for e-commerce platforms) who make monetary transfers to these app companies in return. While standard economic theory [Myerson, 1981; Mas-Colell *et al.*, 1995; Milgrom, 2004; Shoham and Leyton-Brown, 2009] provide good mechanism frameworks for these companies to begin with, they typically only work under very idealized environments, where the participants are perfectly rational, play only once, and their private information is single-dimensional, statistically known or doesn't change over time, etc. None of these assumptions hold in the practices of these nationwide mobile apps, where players may have different levels of rationality, come to and leave the app over time, and have private information that may also change over time.

As a result, what the classic theory provides to these company is merely one mechanism in some, usually parameterized, class and these companies all deploy a team of engineers and scientists to constantly tune and optimize within the class so as to accommodate dynamical information reflected in the huge amount of data generated by daily executions of the mechanisms[1]. Such manual optimizations turn out to be useful in the sense that tiny improvements in a single mechanism can lead to huge revenue gain due to large scale and high execution frequency. On the other hand, they also waste costly human resources and can be erroneous and ad hoc at times. A more sophisticated procedure that can automatically incorporate dynamic information and optimize mechanism parameters should be in place.

1.1 High Frequency Mechanism Design

In this paper, we aim to design automated meta-mechanisms that produce optimized mechanisms in the dynamic environments. These environments, coined *high frequency mechanism design* environments, share the following features:

- There is a large, sometimes variable, set of players;
- Players are strategic to some extent, however, information that affects players' decisions may be complex, unknown and changes dynamically over time. Moreover, different players may have different levels of rationality;

*This work was supported in part by the National Natural Science Foundation of China Grant 61561146398, a Tsinghua University Initiative Scientific Research Grant and a China Youth 1000-talent program. The author is grateful to Qingpeng Cai, Aris Filos-Ratsikas and Weiran Shen for helpful discussions.

[1] See a field experiment conducted by Yahoo! on manually testing and optimizing reserves prices on keyword auctions [Ostrovsky and Schwarz, 2011]

- Players partially observe the mechanism parameters and the outcome, or feedbacks that are relevant to them;
- Mechanisms are executed at high frequencies;
- The designer is flexible to adjust parameters of the mechanism within some predefined class;
- The designer has massive data from past executions of the mechanisms, however, all the data are typically generated from few mechanisms in the class;
- The designer is interested in long-term objectives such as cumulative (discounted) revenue over the next couple of months.

Examples abound. Typical environments include that of advertisement auctions on search engines where auctions happen hundreds of times per second and search engines get to dynamically adjust the reserve prices under the generalized second price (GSP) auction class [Edelman *et al.*, 2007] or the "squash" class [Lahaie and Pennock, 2007], and observe revenue generated afterwards. They also include designing ranking mechanisms for electronic commerce sites where a ranking of sellers (and their related products) will be calculated whenever a buyer query is entered, typically hundreds of times per second. The sites dynamically adjust ranking mechanisms (and thereby adjust allocations of buyer impressions) and observe the revenue, or total transaction volume generated afterwards [Cai *et al.*, 2016]. They also include scenarios of online ride-sharing platforms where matchings and pricing need to be calculated every few seconds and the platform gets to adjust their matching and pricing mechanisms dynamically and observe the revenue or GMVs (gross merchandise volumes) afterwards.

1.2 Challenges

Non-standard environments described above raise a few challenges that seem hard to reconcile when one tries to adopt standard theories and techniques in the field.

Challenges to game theory and mechanism design

There are a few perhaps obvious challenges to the theory and practice of mechanism design. First of all, there is no clear utility model for players of this kind. If one were to model players using Bayesian games, one clear challenge is that there is no explicit definition of type in these application domains. Each player's decision is affected by multiple factors (sometimes without physical meanings) that lie implicitly in the data. Even if one were able to extract these factors, when it comes to revenue, despite considerable efforts from the community, the nature of the optimal mechanism is still not well understood [Tang and Sandholm, 2012; Hart and Nisan, 2012; Cai *et al.*, 2012; Yao, 2015; Tang and Wang, 2016; Mirrokni *et al.*, 2016]. Furthermore, due to informational and computational constraints, the players are not fully rational. Finally, the dynamic nature of the problem further excludes surviving candidates from existing theories.

Challenges to machine learning

Difficulties in modeling may make one wonder the possibility to use a data-driven approach. That is, to use the data generated in the process to learn the player's type and hence the utility model. Indeed, this is the approach adopted in a recent attempt to model players in sponsored search auctions via no-regret learning [Nekipelov *et al.*, 2015] where the authors try to infer advertisers' valuations from bidding data under the GSP framework, under the assumption that the bidders must use a strategy that yields zero regret (as opposed to best response) in the long run. They still assume each bidder has a single dimensional type and they are rational in that they use a no-regret strategy. In addition, their results yield a range of valuations of advertisers that fit the data, not exact values.

Perhaps an even more challenging obstacle is the lack of variety in data. That is, the designer only has data generated by mechanisms defined by only few sets of parameters. This is typically due to lack of exploration in the past. It is theoretically challenging to make use of this data to predict the mechanism performance with a similar set of parameters[2].

There are a few papers that take the initiative to tackle the challenging problems above. In the context of revenue optimization in auction design, two recent papers [Mohri and Medina, 2016; 2015] apply learning algorithms to exploit past auctions. Their algorithms focus on estimation of the underlying bid distribution, thus depending on the assumption that players do not change their behavior over time. Another line of work [Mohri and Munoz, 2015; 2014] aims to maximize revenue with strategic buyers who aim to maximize their cumulative discounted surplus. They give online pricing algorithms with desirable regret bounds. These works still assume that there exists an underlying bid (value) distribution and the buyers are perfectly rational with respect to this single-dimensional type.

In the sponsored search context, two related papers [He *et al.*, 2013; Tian *et al.*, 2014] assume that buyers are Markovian in that their decisions only depend on observations from the previous day and try to find the optimal static mechanism, as opposed to our goal of finding the optimal dynamic mechanisms. They also restrict their player model to be a linear combination of several simple behavior patterns. In addition, they seem to mitigate the game-theoretical effects for different parameters of the mechanism by assuming a uniform markov model across all games.

2 Reinforcement Mechanism Design

In this section, we describe a framework to tackle the high frequency mechanism design problem. We first present, with certain degree of abstraction, the structure of our framework and then present two instantiations of this framework with implementation details under representative application scenarios, one with the ranking mechanism design setting with Taobao [Cai *et al.*, 2017] and the other with the sponsored search auction design setting with Baidu [Shen *et al.*, 2017].

2.1 Dynamic Mechanism Design as an MDP

Our key insight is to model the high frequency mechanism design problem as a policy finding problem in a related *Markov*

[2]For example, it is interesting whether players' behavior (say, Nash equilibrium strategies) possesses continuity when the game under consideration continuously moves in the parameter space.

decision process (MDP), where a *state* encodes all historical action profiles and outcomes produced by mechanisms in past rounds[3]; an *action* is a set of parameters that defines a mechanism in the class; an *immediate reward* (say, revenue of this round) is defined to be a function of the outcome of this round; a *state transition* in this case is simply to append the action and outcome data to the previous state.

In other words, in the current state where the designer possesses all the historical data so far, he takes an action and hence chooses a new mechanism from the parameterized class. The players observe, either fully or partially; directly or indirectly, the parameters, take into account their local information and then react strategically to the new mechanism. At the end of the day, the designer observes the outcome as well as the immediate reward associated with the outcome. The global state then progresses to the next day by appending the data generated today to the current state while each player also locally progresses to the next day by incorporating information he or she observes today. One can easily verify that the definition above satisfies the Markov property.

The designer's goal is to find a policy, one action for each state on the planning trajectory, that enjoys desirable cumulative discounted reward.

2.2 Players Model

The basic idea here is to model each player as an independent[4], local Markov decision process, where a local state encodes the part of historical actions and outcomes that the player can observe so far (again, we can also model a local state of a player to be the set of records observable of the past k rounds and the Markov property persists); an *action* in the local MDP is exactly a feasible action defined by the mechanism; the player then observes the partial outcome on this state and derives his or her utility as immediate reward in the MDP; the player incorporates new data observed, updates his local records and the local state is updated accordingly.

Consequently, a strategy (or policy) of a player is a function that maps each local state to a distribution of actions.

An alternative way to model the mechanism-player interaction above is by general-sum Markov games (see e.g., [Leibo *et al.*, 2017] where each local state is modeled as an observation. We comment that, when it comes to design optimization algorithms, this alternative optimization does not seem to give much technical convenience.

2.3 Solving the Designer's MDP

There are a few facts about the above MDP that makes it nontrivial to solve. First of all, the number of states scales exponentially with respect to the number of players. For realistic applications, the number of players can easily be several hundreds. So standard methods such as value and policy iterations are unlikely to work in such settings. Secondly, states and immediate rewards are generated and observed online (i.e., unknown in advance), so optimization algorithms need to explore for sufficient rounds to gather reward information, which slows down convergence. Last but not least, the action space is sometimes continuous, which further slows down convergence. In the following sections, we design optimization algorithms, tailored for each problem domain, that solve or circumvent these difficulties.

[3] In implementation, one can view a state as the set of records of the past k rounds, with fixed k. The Markov property still persists.

[4] Here, "independence" is in the same spirit as elaborated in [Leibo *et al.*, 2017]: each individual's decision processes are independent of one another in the sense that each regards the others as a part of the environment. As a result, the others' strategies affect this agent only implicitly by affecting the environment of this agent. One can regard this assumption as a form of bounded rationality where players do not explicitly reason and react to what others do, but reason from what the player can observe at its local state.

3 Case Study I: Buyer Impression Allocations in E-commerce Platforms

As a joint project with Taobao [Cai *et al.*, 2017], we instantiate our framework on the domain of electronic commerce where platforms allocate buyer impressions/visits to sellers, aiming to maximize the total revenue generated by the platform. When a buyer types a keyword query (normally hundreds of times per second), the website returns to the buyer with a ranking list of sellers for this item, together with the corresponding prices. Different rankings correspond to different allocations of click through rates (CTRs), so that the above process can be regarded as an instance of repeated resource allocation problem in which the sellers choose their prices at each round and the platform decides how to allocate the impressions, based on the chosen prices and the historical transactions of each seller.

3.1 MDP Formulation

Formulated using the language in Section 2, a state of the corresponding MDP consists of the records (action, allocation outcome, etc) of all sellers in the last k rounds; an action in this case is for the platform to choose an allocation of impressions, i.e., a division of each unit of buyer impression to all sellers related to the buyer query; the immediate reward to the platform is the total expected revenue generated in this round; and the new state is simply the sellers records of the last k rounds, looking back from the next day. Performance of an allocation algorithm is evaluated by the average expected total revenue over the next k_0 rounds.

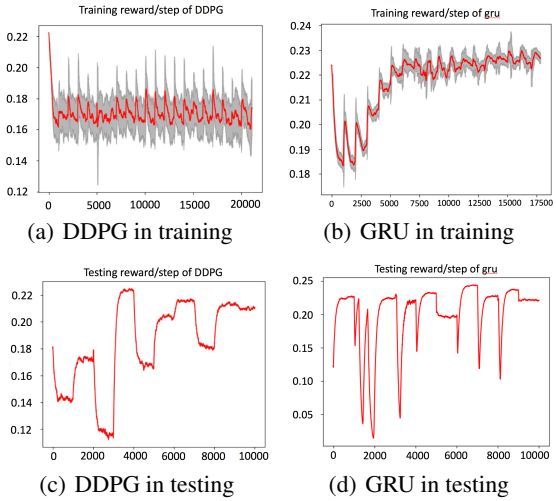
3.2 Sellers' Behavior Model

Given a price p set by a seller, the probability that the item is purchased is given by $\lambda(1 - F(p))$ where $F(p)$ estimates the probability that an average buyer's valuation is below p and λ is the fraction of impression allocated to this seller. In [Cai *et al.*, 2017], we generalize this probability to the case where a buyer's purchase probability is also dependent on the seller's historical transaction record.

Given buyers' behavior, a seller's model is as follows:

At round 0, each seller posts a random price. At any other round $\tau + 1$ (with $\tau \geq 0$), for some $\epsilon_i \in [0, 1]$,

- with probability $1-\epsilon_i$, seller i selects a price drawn from the empirical distribution of her historical price choices in the previous k rounds;
- with probability ϵ_i, seller i picks the price of round τ_0 with the maximum discounted profit in the last k rounds



(a) DDPG in training (b) GRU in training

(c) DDPG in testing (d) GRU in testing

Figure 1: Rewards/round of algorithms in training and testing

and furthermore adds Gaussian noise for exploration; that is, the seller posts a price $p_{i\tau_0} + a$ at round $\tau + 1$, where a is the Gaussian noise. The use of the discount factor follows the standard assumption that recent records will factor more in the decision of a seller.

According to existing data, the model above fits well with sellers' behavior in Taobao. A similar yet different idea has been proposed lately [Ortega and Stocker, 2016] to model rationality-statistics tradeoff in repeated games.

3.3 Optimization via Deep Reinforcement Learning

As argued in the previous section, the MDP has continuous action spaces and exponential many states in the number of sellers. To handle the continuous action space, we borrow insights from the *Deep Deterministic Policy Gradient* algorithm [Lillicrap *et al.*, 2015]. To handle the huge number of states, we decompose the original neural network into a set of sub-networks, one of each seller, to make use of the independence property mentioned in footnote 4 as well as the factor that the Q value (aka. state-action value) of an action (an allocation) is the sum of the Q values of each individual allocation. As a result, our algorithm scales linearly in the number of sellers and converges on large realistic instances.

3.4 Experiments

We compare our algorithm (called GRU) with the DDPG algorithm and several other algorithms, the readers are referred to [Cai *et al.*, 2017] for a full description of the experimental results. We list in Figure 1 the comparison with DDPG. It can be seen that the (normalized) average reward of our algorithm is higher in both testing and training data. The reason that there are fluctuations in the testing data is that we sample sellers' cost functions from a distribution every few rounds to simulate the dynamic environment and thus both algorithms need several rounds to adjust to the new cost functions.

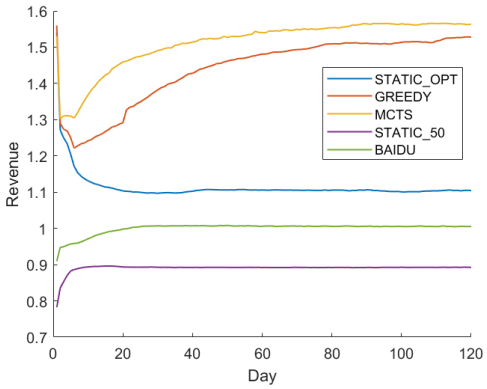


Figure 2: Comparison of reserve pricing schemes

4 Case Study II: Dynamic Reserve Pricing in Sponsored Search Auctions

As a joint project with Baidu [Shen *et al.*, 2017], the largest search engine in China, we instantiate our framework to optimize reserve prices in sponsored search auctions[5].

Using the formulation of Section 2, the search engine's state is described by the bidding data and past allocations and payments, while an action in each round is to design a new sets of reserve prices, one for each advertiser. The immediate reward is the per round revenue generated by the search engine.

In this project, instead of explicitly modeling each advertiser's bidding strategy as what we did in the case of Taobao, we use a deep neural network (LSTM) to adaptively learn from its past bidding data and feedback, and predict its future bid distribution. To optimize the designer's MDP, we first discretize the action space and then make use of the Monte-Carlo tree search (MCTS) algorithm to speed up the forward-looking search. Readers are referred to [Shen *et al.*, 2017] for the implementation details.

We compare our approach with several dynamic reserve schemes as well as static ones, including the one that is currently deployed by Baidu, using a 8 month real bidding dataset of 400 keywords. The results are listed in Figure 2.

Our algorithm (MCTS) yields the highest revenue at convergence. Another dynamic algorithm called GREEDY, that in each round selects one with the largest gradient of revenue, also performs very well. Dynamic pricing schemes outperform all static schemes with large margins, including the one (BAIDU in the figure) that is currently deployed.

5 Concluding Remarks

By making use of recent advances in AI, we demonstrate that one can automatically design and optimize economic mechanisms at nationwide scales for important industrial applications. We are currently experimenting with DiDi, the largest ride-sharing app in China, to further validate our framework.

[5]In a parallel work, we implement our approach to optimize ranking rules in sponsored search auctions [Shen and Tang, 2017].

References

[Cai *et al.*, 2012] Yang Cai, Constantinos Daskalakis, and S. Matthew Weinberg. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *FOCS*, pages 130–139, 2012.

[Cai *et al.*, 2016] Qingpeng Cai, Aris Filos-Ratsikas, Chang Liu, and Pingzhong Tang. Mechanism design for personalized recommender systems. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pages 159–166, 2016.

[Cai *et al.*, 2017] Qingpeng Cai, Aris Filos-Ratsikas, Pingzhong Tang, and Yiwei Zhang. A deep reinforcement learning framework for allocating buyer impressions in e-commerce websites. *Working paper*, 2017.

[Edelman *et al.*, 2007] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American economic review*, 97(1), 2007.

[Hart and Nisan, 2012] Sergiu Hart and Noam Nisan. Approximate revenue maximization with multiple items. In *ACM Conference on Electronic Commerce*, page 656, 2012.

[He *et al.*, 2013] Di He, Wei Chen, Liwei Wang, and Tie-Yan Liu. A game-theoretic machine learning approach for revenue maximization in sponsored search. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[Lahaie and Pennock, 2007] Sébastien Lahaie and David M. Pennock. Revenue analysis of a family of ranking rules for keyword auctions. In *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11-15, 2007*, pages 50–56, 2007.

[Leibo *et al.*, 2017] Joel Z. Leibo, Vinícius Flores Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 464–473, 2017.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Mas-Colell *et al.*, 1995] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, June 1995.

[Milgrom, 2004] Paul Milgrom. *Putting Auction Theory to Work*. Cambridge University Press, 2004.

[Mirrokni *et al.*, 2016] Vahab S. Mirrokni, Renato Paes Leme, Pingzhong Tang, and Song Zuo. Dynamic auctions with bank accounts. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 387–393, 2016.

[Mohri and Medina, 2015] Mehryar Mohri and Andres Muñoz Medina. Non-parametric revenue optimization for generalized second price auctions. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*, pages 612–621, 2015.

[Mohri and Medina, 2016] Mehryar Mohri and Andrés Munoz Medina. Learning algorithms for second-price auctions with reserve. *Journal of Machine Learning Research*, 17(74):1–25, 2016.

[Mohri and Munoz, 2014] Mehryar Mohri and Andres Munoz. Optimal regret minimization in posted-price auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, pages 1871–1879, 2014.

[Mohri and Munoz, 2015] Mehryar Mohri and Andres Munoz. Revenue optimization against strategic buyers. In *Advances in Neural Information Processing Systems*, pages 2530–2538, 2015.

[Myerson, 1981] Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[Nekipelov *et al.*, 2015] Denis Nekipelov, Vasilis Syrgkanis, and Éva Tardos. Econometrics for learning agents. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 1–18, 2015.

[Ortega and Stocker, 2016] Pedro A Ortega and Alan A Stocker. Human decision-making under limited time. In *Advances in Neural Information Processing Systems*, pages 100–108, 2016.

[Ostrovsky and Schwarz, 2011] Michael Ostrovsky and Michael Schwarz. Reserve prices in internet advertising auctions: a field experiment. In *Proceedings 12th ACM Conference on Electronic Commerce (EC-2011), San Jose, CA, USA, June 5-9, 2011*, pages 59–60, 2011.

[Shen and Tang, 2017] Weiran Shen and Pingzhong Tang. Practical versus optimal mechanisms. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 78–86. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[Shen *et al.*, 2017] Weiran Shen, Binghui Peng, Hanpeng Liu, Michael Zhang, Ruohan Qian, Yan Hong, Zhi Guo, Zongyao Ding, Pengjun Lu, and Pingzhong Tang. Reinforcement mechanism design, with applications to dynamic pricing in sponsored search auctions. Working paper, 2017.

[Shoham and Leyton-Brown, 2009] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game theoretic and Logical Fundations.* Cambridge Uni. Press, 2009.

[Tang and Sandholm, 2012] Pingzhong Tang and Tuomas Sandholm. Mixed bundling auctions with reserve prices. In *AAMAS*, 2012.

[Tang and Wang, 2016] Pingzhong Tang and Zihe Wang. Optimal auctions for negatively correlated items. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 103–120, 2016.

[Tian *et al.*, 2014] Fei Tian, Haifang Li, Wei Chen, Tao Qin, Enhong Chen, and Tie-Yan Liu. Agent behavior prediction and its generalization analysis. *arXiv preprint arXiv:1404.4960*, 2014.

[Yao, 2015] Andrew Chi-Chih Yao. An n-to-1 bidder reduction for multi-item auctions and its applications. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 92–109, 2015.