# Learning Stackelberg Equilibria in Sequential Price Mechanisms

**Gianluca Brero** [1] **Darshan Chakrabarti** [1] **Alon Eden** [1] **Matthias Gerstgrasser** [1] **Vincent Li** [1] **David Parkes** [1]

## Abstract

We study the problem of the design of simple economic mechanisms for assigning items to self-interested agents that combine a messaging round with a sequential-pricing stage. The rules of the sequential-pricing stage and in particular the way these rules use messages determines the way the messaging stage is used. This is a Stackelberg game where the designer is the leader and fixes the mechanism rules, inducing an equilibrium amongst agents (the followers). We model the followers through equilibrium play coming from no-regret learning, and introduce a novel single-agent *Stackelberg MDP formulation*, where the leader learns to effect a follower equilibrium that optimizes its objective. We solve this MDP using actor-critic methods, where the critic is given access to the joint information of all the agents.

## 1. Introduction

Despite the successful application of machine learning for the automated design of *direct mechanism* for allocating items to a multi-agent system of self-interested agents (Duetting et al., 2019, e.g.,), the automated design of *indirect mechanisms* is less well understood. Indirect mechanisms are interesting because they simplify the reports required of agents. In particular, participation does not require an agent to report its complete preferences over allocations (as is required in a direct mechanism).

In this paper, we consider the novel problem of the automated design of indirect mechanisms that combine a messaging round with a simple, sequential-pricing stage. The sequential pricing stage makes a take-it-or-leave-it offer of remaining items to each agent in turn, with an agent having the option to select an item. Agents in this stage have a dominant strategy (take the item, if any, that maximizes its true utility). Both the order and the prices offered can depend on the messages received by the mechanism as well as previous choices by agents.

This family of mechanisms is motivated by different practical applications, including the following:

- Online platforms such as Priceline, that offer limited bidding capability according to which they choose a match, for example of a consumer to a hotel.
- School matching mechanisms, that tend to elicit limited preferences from parents of students. For, instance, the top three school choices among schools in the district. The platforms take these reports into account when implementing a priority-based, round-robin matching algorithm (Center, 2020).
- Online marketplaces such as Amazon offer personalized deals based on reported information; e.g., Amazon Prime members can customize deals based on themes that they select such as "sports enthusiast," and Amazon also encourages customers to make use of the "improve your recommendations" part of a user's profile.

What is interesting is that the rules of the sequential-pricing stage (the *policy* of the mechanism), and in particular the way these rules make use of messages, determines the way the messages are used. That is, the way in which the "listener" (the mechanism) makes use of the messages affects the use to which the "speakers" (the agents) put the messages. In this way, the semantics of the messaging round arise endogenously as a result of the rules of the mechanism.

We model this as a *Stackelberg game* (Von Stackelberg, 2010), where the designer is the *leader*—and fixes the mechanism rules—these rules inducing an equilibrium amongst agents (the *followers*). We model the equilibrium behavior of the followers as coming from *no-regret learning*, and introduce a novel single-agent *Stackelberg MDP formulation*, where the leader learns to effect a follower equilibrium that optimizes the leader's objective. We solve this Stackelberg MDP using actor-critic reinforcement learning, where the critic is given access to the agents' joint information.

The use of no-regret learning dynamics leads the agents' behavior to converge a *Bayesian coarse-correlated equilibrium* (B-CCE) (Hartline, 2002). Our key insight is that,

[1]Author order is alphabetical. John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, United States . Correspondence to: Gianluca Brero <gbrero@g.harvard.edu>.

*Proceedings of the 38$^{th}$ International Conference on Machine Learning*, PMLR 139, 2021. 

since this learning happens through agents perturbing their inputs to the mechanism policy, we can formulate these dynamics as a part of an extended MDP (the Stackelberg MDP). In this way, the mechanism can learn how its policy influences the equilibrium play of agents. We prove that the optimal policy of the Stackelberg MDP forms a Stackelberg equilibrium, solving the mechanism design problem.

This work extends earlier work (Brero et al., 2021), which considered only sequential pricing mechanisms (SPMs) without message passing. Since agents have a dominant strategy in SPM without messaging, this earlier work could solve the Stackelberg learning problem by simply modeling this behavior of agents directly within the leader's MDP. There was no need for the Stackelberg MDP that we introduce here. We provide simulations to show the benefits that come from adding message passing, and suggest the utility of this framework in enabling Stackelberg learning.

**Learning Stackelberg Equilibria** There has been significant interest in learning equilibria for Stackelberg games in the context of security games (Li et al., 2019; Sengupta and Kambhampati, 2020; Xu et al., 2021), where security forces (leaders) commit to some surveillance strategies, and adversaries (followers) best respond to those. Here, positive results on equilibrium convergence are generally related to single-leader single-follower (and often zero-sum) games, conditions not required by our approach.

Another research thread has focused on identifying conditions under which gradient descent dynamics converge to Stackelberg equilibria (Fiez et al., 2019). While the positive results here are only related to zero-sum games and settings with only one leader and one follower, our framework is more general and does not require followers' strategies to be differentiable.

In another related work, Zhang et al. (2020) developed learning dynamics that converge to a set of equilibria that includes Stackelberg ones but require the globally optimal action profile at every state to be chosen at every iteration. Our approach is guaranteed to converge to Stackelberg equilibria and does not make any such assumption regarding the structure of the game (the globally optimum point might not even exist) or choices made by the algorithm.

## 2. Preliminaries

There are $n$ agents and $m$ indivisible items. Let $[n] = \{1,\dots,n\}$ be the set of agents and $[m] = \{1,\dots,m\}$ be the set of items. Each agent $i$ has a valuation function $v_i : 2^{[m]} \to \mathbb{R}_{\geq 0}$ that maps bundles of items to a real value. Let $\mathbf{v} = (v_1,\dots,v_n)$ denote the valuation profile. We assume $\mathbf{v}$ is sampled from a possibly correlated value distribution $\mathcal{D}$. This distribution has support $\bigtimes_{i=1}^{n} \mathcal{V}_i$, where each $\mathcal{V}_i$ is a discrete space called agent $i$'s *type space*. The designer can access the value distribution $\mathcal{D}$ through samples.

An *allocation* $\mathbf{x} = (x_1,\dots,x_n)$ is a profile of disjoint bundles of items ($x_i \cap x_j = \emptyset$ for every $i \neq j \in [n]$), where $x_i \subseteq [m]$ is the set of items allocated to agent $i$. We call the quantity $\sum_{i\in[n]} v_i(x_i)$ *social welfare* of allocation $\mathbf{x}$ under valuation profile $\mathbf{v}$. An *economic mechanism* interacts with agents and determines an allocation $\mathbf{x}$ and transfers (payments) $\boldsymbol{\tau} = (\tau_1,\dots,\tau_n)$, where $\tau_i \geq 0$ is the payment by agent $i$. In this paper we focus on designing mechanisms that maximize the *social welfare* of their final allocation.

**Sequential Price Mechanisms with Messages** We study the class of *message sequential price mechanisms* ($\mu SPM$s). This extends the class of $SPM$s (Brero et al., 2021) by including an initial round of communication between agents and mechanism.

A $\mu SPM$ interacts with agents across rounds. In the first round, each agent $i$ sends a *message* $\mu_i$ chosen from a set of $M_i$ options. Let $\mu = (\mu_1,\dots,\mu_n)$. Each agent $i$ chooses her message following a *messaging strategy* $\sigma_i(v_i)$ that defines a probability distribution over $M_i$. Let $\sigma = (\sigma_1,..,\sigma_n)$.

The mechanism then visits each agent in turn in each of the following rounds. In each round, the mechanism picks an unvisited agent $i$, and posts a price $p_j$ for each available item $j$. Then, agent $i$ picks the bundle of items that maximizes their utility, and is charged with the corresponding item prices.

We show that learning an optimal $\mu SPM$ is equivalent to learning an optimal policy in a suitably defined POMDP. Furthermore, the discrete message space that $\mu SPM$ provides to each agent allows deriving optimal messaging strategies via simple no regret algorithms.

**Solution Concept** In our setting, both the agents and the mechanism are adapting. The mechanism adapts itself to agents' behavior in order to optimize its objective, and the agents adapt their strategies to the new mechanism rules. We are interested in finding a *Stackelberg equilibrium*, where the mechanism is the leader and the agents are the followers.

**Definition 1** ***(Bayesian Stackelberg Equilibrium (Paruchuri et al., 2008))*** *In a Bayesian Stackelberg equilibrium, there is a leader with a fixed type, and followers with types drawn from a distribution. The leader first commits to a strategy, and followers adopt a strategy profile that forms an equilibrium in the game induced by the leader's strategy.*

In this work, we assume the followers play a *Bayesian coarse-correlated equilibrium* (B-CCE). B-CCEs naturally

arise from no regret dynamics.[1] In the next section we formulate the learning that corresponds to the mechanism design problem.

## 3. Learning Multi-agent Mechanisms as a Single Agent Problem

Consider a *partially observable stochastic game* (Hansen et al., 2004) among the $n$ agents and the mechanism. The *game state* $s^t = (\mathbf{v}, \mu, \mathbf{x}^{t-1}, \rho^{t-1})$ is a tuple consisting of the valuation profile $\mathbf{v}$, the (initially empty) message profile $\mu$, the current partial allocation $\mathbf{x}^{t-1}$, and the residual setting $\rho^{t-1}$ consisting of agents not yet visited and items not yet allocated. The full state of the game is not observed directly, but each agent observes partial information about each state the game is in. At round 0, each agent observes its own value and sends a message. At round 1, the mechanism *observes* these messages ($o^1 = \mu$). At any round $t \geq 1$, the mechanism takes *action* $a^t = (i^t, p^t)$, where $i^t$ is the next selected agent and $p^t$ is the vector of posted prices offered in that round. Agent $i^t$ chooses a utility maximizing set of items $x^t$ (perhaps empty) at prices $p^t$, which is observed by the mechanism[2] ($o^t = x^{t-1}$ for any $t \in \{2, .., T\}$).

The first state transition just adds agents' messages to the state. Then, at any round $t > 0$, the state $s^{t+1}$ is obtained by adding the bundle $x^t$ selected by agent $i^t$ to the partial allocation $\mathbf{x}^{t-1}$ to form a new partial allocation $\mathbf{x}^t$, and the items and agent are removed from the residual setting $\rho^{t-1}$ to form $\rho^t$. The mechanism's *reward* $r(s, a)$ is zero in all states except for terminal states, defined to be states in which no agents or items are left. This reward can capture any objective of the designer. In this work, we consider *social welfare*: the reward is the total realized agent value.

Since agent strategies are a function of the mechanism's policy, we can write the optimization problem as an optimization over the policy's parameters $\theta$. Let $\sigma_\theta$ be an equilibrium induced by a mechanism (policy) represented by parameters $\theta$. Also, let $\text{tr}^{\text{SG}} = (\mathbf{v}, \mu, s^1, a^1, \ldots, a^T, s^T)$ be a *trajectory* of an episode of the stochastic game.

The objective is to find parameters $\theta$ that maximizes $J(\theta) = E_{\text{tr}^{\text{SG}} \sim p_\theta}\left[\sum_{t=0}^{T} r(s^t, a^t)\right]$, where

$$p_\theta(\text{tr}^{\text{SG}}) \;=\; p(\mathbf{v})\,\sigma_\theta(\mu \mid \mathbf{v}) \prod_{t=1}^{T} \pi_\theta(a^t|o^t) p(s^{t+1}|s^t, a^t). \quad (1)$$

This is the distribution over trajectories that is induced by the choice of mechanism policy $\theta$. Observe that the optimal solution to this optimization problem is a Bayesian Stackelberg equilibrium, where the mechanism is the leader, and the agents act as followers.

As in Brero et al. (2021), we want learn policy $\pi_\theta$ using *policy gradient* methods. Notice that the stochastic game defined above is not a proper MDP, as the messages that the agents bid depend on the policy itself. This introduces new challenges, where we want to have the mechanism be aware of how the probability distribution of bids changes in response to its policy. We address this by using the policy to determine the agents' equilibrium strategies within the episode. To model the equilibrium messaging of agents we use *no-regret dynamics*. In no-regret dynamics, agents update their strategies by examining their utility in the mechanism outcome under different possible messages. For this reason, this can be computed by rolling out the policy itself. The crucial idea is that we can expand $\sigma_\theta$ to include the initial part of an MDP that calls the policy multiple times to compute agents' strategies with a no-regret algorithm. By doing this, we expose the policy to its role in determining agents' strategies. Thus, we can apply RL methods to optimize over the parameters of the policy for this MDP.

**The Stackelberg MDP** Our MDP, which we name the *Stackelberg MDP*, has a long episode that consists of sub-episodes. A sub-episode consists of rolling out the policy to determine a allocation and payments. We have two types of sub-episodes:

1. *Equilibrium sub-episodes:* The first set of sub-episodes are used by the agents to find the equilibrium of the message-passing game, where for some $T$ rounds of no-regret dynamics:
   (a) Agents' values are drawn.
   (b) Agents jointly sample their messages according to their current strategy.
   (c) Each agent runs an algorithm that minimizes its external-regret[3] in the full information setting: it computes the utility it would have received for every possible message sent, fixing all other agents' messages. The agent then uses this utility vector to update its strategy using a no-regret algorithm.
2. *Reward sub-episode:* After running $T$ equilibrium sub-episodes, we run a sub-episode where the policy gets a reward for agents playing the current messaging strategy (as determined in the first $T$ rounds). The reward of the final sub-episode is the reward of the entire episode.

To maintain Markov properties, we include the quantities that determine how no regret dynamics evolve in the state

[1] We defer the formal definition of B-CCE and the general proof that no regret dynamics converge to B-CCEs to the supplementary material.

[2] We assume that agents play their dominant strategy for this part of the game.

[3] External regret compares the performance of a sequence of actions to the performance of the best single action in hindsight.

space. In the following subsection, we show how we can derive a mechanism that only uses agents' messages and purchases to determine its actions, despite being trained with this extra information.

In the supplementary materials, we show that the first type of sub-episodes converges to a Bayesian Coarse-Correlated equilibrium. This implies the following corollary.

**Proposition 1** *As $T$ goes large, the optimal policy of the Stackelberg MDP gets the same utility as the leader in the Stackelberg game between the mechanism and the bidders.*

**Proof 1** *Notice that fixing agents valuations and bids, the second phase of the Stackelberg MDP is identical to round 1 onward of the partially observable stochastic game, where the policy induced by $\theta$ is used to allocate the items to the agents, and gets the appropriate reward. Let $\theta^*_T$ be the optimal policy for the Stackelberg MDP for $T$ first phase steps, and $\sigma_{\theta^*_T}$ the strategies implied by running the first phase with policy $\theta^*_T$. As $T$ grows large, $\sigma_{\theta^*_T}$ converges to a B-CCE as shown in the supplementary material. By optimality of $\theta^*_T$, there is no policy that could achieve better by switching to $\theta'$ and and letting agents play according to the new equilibrium $\sigma_{\theta'}$. Therefore, $(\theta^*_T*, \sigma_{\theta^*_T})$ converge to a Bayesian Stackelberg Equilibrium.*

**An Actor-critic Approach** Through the Stackelberg MDP, the entire optimization problem is a single-agent MDP, where the only actions are the actions of the mechanism's policy. However, when deployed, the mechanism won't have access to the agents' internal equilibrium computation, but only to the observations $o^t$ consisting of the agents' messages and purchases. We train the mechanism's policy by adopting an actor-critic approach. We first define a generic trajectory $\mathsf{tr}^{\mathsf{MDP}} = (s^0, a^0, \dots, s^T, a^T)$ in our Stackelberg MDP. We can express the trajectory probability of the optimization problem as

$$p_\theta(\mathsf{tr}^{\mathsf{MDP}}) = p(s^0)\prod_{t=0}^{T}\pi_\theta(a^t|o^t)p(s^{t+1}|s^t,a^t),$$

where $p(s^0)$ now doesn't depend on $\theta$, and the states and observations are defined by the Stackelberg MDP.

The gradient of $J(\theta)$ with respect to $\theta$ can be expressed as

$$\begin{aligned}\nabla_\theta J(\theta) &= \mathbb{E}_{\mathsf{tr}\sim p_\theta}\left[\nabla_\theta \log p_\theta(\mathsf{tr})\left(\sum_{t'=1}^{T} r(s^{t'},a^{t'})\right)\right]\\ &= \mathbb{E}_{\mathsf{tr}\sim p_\theta}\left[\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a^t|o^t)\left(\sum_{t'=1}^{T} r(s^{t'},a^{t'})\right)\right]\\ &= \mathbb{E}_{\mathsf{tr}\sim p_\theta}\left[\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a^t|o^t)\left(\sum_{t'=t}^{T} r(s^{t'},a^{t'})\right)\right]\end{aligned}$$

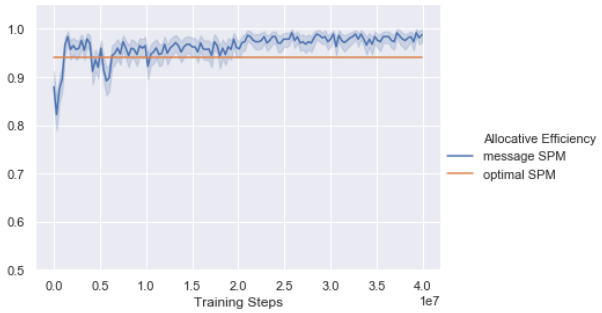


*Figure 1.* Performance as a function of training steps of an $\mu SPM$.

where the last equality follows since future actions do not affect past rewards in an MDP. $\nabla_\theta J(\theta)$ is approximated by sampling $\ell$ different trajectories $\mathsf{tr}_1, .., \mathsf{tr}_\ell$:

$$\nabla_\theta J(\theta) \approx \frac{1}{\ell}\sum_{k=1}^{\ell}\sum_{t=1}^{T^k}\nabla_\theta \log \pi_\theta(a^t_k|o^t_k)\left(\sum_{t'=t}^{T^k} r(s^{t'}_k, a^{t'}_k)\right).$$

We reduce the variance of this gradient by replacing the term $\sum_{t'=t}^{T} r(s^{t'}_k, a^{t'}_k)$ with $Q^\theta(s^t_k, a^t_k)$, where $Q^\theta(s,a)$ is the standard critic network with access to the full MDP state (including valuations, strategies, etc.) which is inaccessible to the policy.[4] This approach based on centralized training and decentralized execution is similar to the one proposed for DDPG by Lowe et al. (2017).

Note that the actor-critic approach described above allows us to maintain our POMDP tractable, as we do not need to provide a sufficient statistic of the history of observations as input to the policy (unlike Brero et al. (2021)). Indeed, standard actor-critic algorithms only require the environment to be Markovian for the critic network. Given that this network has access to agents' valuations, it does not need to infer them via the history of transactions.

## 4. Illustrative Experimental Results

In this paper we provide only illustrative results that suggest the opportunity to use the Stackelberg MDP framework together with no-regret learning dynamics by agents for the design of indirect mechanisms with message passing.

We test the setting introduced by Agrawal et al. (2020) in their Example 1. We have 2 agents and 1 item. Agent 1's value $v_1$ has support $\{\frac{1}{2}, \frac{1}{2\varepsilon}\}$ with probabilities $\{1-\varepsilon, \varepsilon\}$, and agent 2's value $v_2$ has support $\{0, 1\}$ with probabilities $\{\frac{1}{2}, \frac{1}{2}\}$. Note that the welfare-optimal allocation cannot always be realized via an $SPM$.[5] However, there exists at least one optimal $\mu SPM$ where agents are properly incentivized to

[4] Note that $Q^\theta(s,a)$ can be accessed at training time as we have access to the full state of the MDP.

[5] The welfare-optimal $SPM$ visits agent 2 first and then agent

communicate their types (supplementary material).

In our experiments we use $\varepsilon = 0.2$. We train our $\mu SPM$s via a standard PPO algorithm (Schulman et al., 2017) as implemented by OpenAI Baselines (Hill et al., 2018). We run the PPO algorithm for 40M steps and we evaluate the performance of the system periodically during training using fresh samples drawn in an evaluation environment.

Figure 1 shows the $\mu SPM$ learning curve obtained by averaging the best three out of six random seeds. To better highlight learning performance, we normalize each episode's welfare (Allocative Efficiency). As we can see, we can learn $\mu SPM$s that outperform optimal $SPM$.

## 5. Conclusion

We have provided a new framework for learning indirect mechanisms that allow for communication between agents and the mechanism. The framework allows for the optimal Stackelberg design to be learned, and is demonstrated here in application to learning optimal mechanisms from a family of mechanisms that involve an initial round of communication followed by sequential offers to agents. We achieve this by bringing the learning dynamics of agents into the view of the learning problem of the designer. We provide experiments to demonstrate the effectiveness of this approach in achieving improved performance compared to $SPM$s.

An interesting next step left by our work deals with the ability to learn a mechanism that interacts with agents multiple times, where messages convey interim information about the agents given the current allocation. On the theory side, a next question is under which conditions does a no-regret dynamic converge to a BNE, as we see this is the solution concept we converge to in practice. Moreover, it will be interesting to show concrete bounds on the sample complexity of learning the optimal $\mu SPM$.

1, using price zero in both cases. Here, the optimal allocation is not implemented when $v_1 = 1/2$ and $v_2 = 1/(2\varepsilon)$.

## References


Shipra Agrawal, Jay Sethuraman, and Xingyu Zhang. On optimal ordering in the optimal stopping problem. In *Proc. EC '20: The 21st ACM Conference on Economics and Computation*, pages 187–188, 2020.

Gianluca Brero, Alon Eden, Matthias Gerstgrasser, David C. Parkes, and Duncan Rheingans-Yoo. Reinforcement learning of sequential price mechanisms. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, pages 5219–5227, 2021.

Cambridge Public Schools Student Registration Center. How the junior kindergarten/kindergarten january 2021 lottery works: A detailed guide for families, 2020. URL https://www.cpsd.us/UserFiles/Servers/Server_3042785/File/departments/administration/frc/Kindergarten_Guide_for_Parents%20_2021.pdf.

Paul Duetting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 1706–1715, 2019.

Tanner Fiez, Benjamin Chasnov, and Lillian J Ratliff. Convergence of learning dynamics in stackelberg games. *arXiv preprint arXiv:1906.01217*, 2019.

Eric A Hansen, Daniel S Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th national conference on Artifical intelligence (AAAI)*, volume 4, pages 709–715, 2004.

Jason D Hartline. Dynamic posted price mechanisms. Technical report, Northwestern University, 2002.

Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. https://github.com/hill-a/stable-baselines, 2018.

Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. Robust Multi-Agent Reinforcement Learning via Minimax Deep Deterministic Policy Gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:4213–4220, July 2019. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v33i01.33014213. URL http://www.aaai.org/ojs/index.php/AAAI/article/view/4327.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pages 6379–6390, 2017.

Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Playing games for security: an efficient exact algorithm for solving bayesian stackelberg games. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 895–902, 2008.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sailik Sengupta and Subbarao Kambhampati. Multi-agent Reinforcement Learning in Bayesian Stackelberg Markov Games for Adaptive Moving Target Defense. *arXiv:2007.10457 [cs]*, July 2020. URL http://arxiv.org/abs/2007.10457. arXiv: 2007.10457.

Heinrich Von Stackelberg. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, and Milind Tambe. Robust reinforcement learning under minimax regret for green security. In *Proc. 37th Conference on Uncertainty in Artifical Intelligence (UAI-21)*, 2021.

Haifeng Zhang, Weizhe Chen, Zeren Huang, Minne Li, Yaodong Yang, Weinan Zhang, and Jun Wang. Bi-level actor-critic for multi-agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7325–7332, 2020.