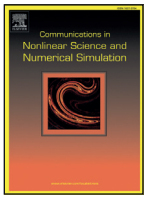
Commun Nonlinear Sci Numer Simulat 99 (2021) 105805

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

journal homepage: www.elsevier.com/locate/cnsns



Research paper

Nash Q-learning agents in Hotelling's model: Reestablishing equilibrium

Jan Vainer[a], Jiri Kukacka[b,c,*]

[a] Charles University, Faculty of Mathematics and Physics, Ke Karlovu 3, Praha 2 121 16, Czechia
[b] Charles University, Faculty of Social Sciences, Institute of Economic Studies, Opletalova 26, Prague 1 110 00, Czechia
[c] Czech Academy of Sciences, Institute of Information Theory and Automation, Pod Vodarenskou vezi 4, Prague 8 182 00, Czechia

ARTICLE INFO

Article history:
Received 22 July 2020
Revised 9 February 2021
Accepted 8 March 2021
Available online 10 March 2021

JEL classification:
C61
C63
C72
L13
R30

Keywords:
Hotelling's location model
Agent-based simulation
Reinforcement learning
Nash Q-learning

ABSTRACT

This paper examines adaptive agents' behavior in a stochastic dynamic version of the Hotelling's location model. We conduct an agent-based numerical simulation under the Hotelling's setting with two agents who use the Nash Q-learning mechanism for adaptation. This allows exploring what alternations this technique brings compared to the original analytic solution of the famous static game-theoretic model with strong assumptions imposed on players. We discover that under the Nash Q-learning and quadratic consumer cost function, agents with high enough valuation of future profits learn behavior similar to aggressive market strategy. Both agents make similar products and lead a price war to eliminate their opponent from the market. This behavior closely resembles the Principle of Minimum Differentiation from Hotelling's original paper with linear consumer costs. However, the quadratic consumer cost function would otherwise result in the maximum differentiation of production in the original model. Thus, the Principle of Minimum Differentiation can be justified based on repeated interactions of the agents and long-run optimization.

Introduction

Some of the most influential economic models stand on game theory. Game-theoretic concepts have been famously used in the past to show that in a society of self-interested individuals, the tragedy of the commons arises [15], or that non-dictatorial voting methods are subject to strategic voting [7]. Game theory allows us to formulate and analyze problems that include decision making in competitive or cooperative environments and offers solution concepts such as the Nash equilibrium. Based on game-theoretic models, we can make conditional conclusions about the behavior of real economic actors. Such models often rely on strong assumptions such as agents' perfect rationality and complete and perfect information. However, humans are neither perfectly rational nor have perfect and complete information available. Thus, in addition to finding the Nash equilibrium in games, we should in the first place also ask how and if ever boundedly rational agents or agents do so without perfect information and get to play the Nash equilibrium.

Experimental economics provides a few studies that partly tackle these issues. In experiments presented, participants were supposed to play certain games multiple times. It has been shown that over time most of the experiment participants

* Corresponding author.
E-mail addresses: vainerjan@gmail.com (J. Vainer), jiri.kukacka@fsv.cuni.cz (J. Kukacka).

https://doi.org/10.1016/j.cnsns.2021.105805
1007-5704/

were getting closer to the Nash equilibrium in the Beauty Contest game [17] and similar conclusions were obtained for bargaining games [22]. These results indicate that some underlying adaptive processes could, at least some games, enable agents to converge towards the Nash equilibrium during repeated interactions. Recently, [2] build a dynamical model of experimental oligopoly games with the Cournot-Nash outcome as a stationary state of the model with two types of agents: adaptive agents that adjust their behavior to increases their profit and agents with imitative behavior. The authors suggest that their model is capable of reproducing the outcomes of experimental oligopoly games qualitatively.

With the rise in modern computers' operational capacity, new techniques for analyzing economic systems have emerged. For instance, using numerical simulations, [21] study a Cournot duopoly model with heterogeneous competitors using the bifurcation analysis and further analyzing the stability switching curves. The authors suggest stability conditions of the unique Nash equilibrium and conclude the stability of the economy. Another such technique is called 'agent-based simulation' that consists of software agents placed in a virtual environment and the environment itself. Agents interact with each other and/or with the environment, and from their micro behavior, a global behavioral pattern can emerge. The rules that guide the agents' behavior range from simple heuristics to more complex, possibly adaptive ones. For example, Waltman and Kaymak [24] use Q-earning to model firms in repeated Cournot oligopoly game, and [12] study differentiated market dynamics for agents imitating the behavior of more successful agents. Nagel and Vriend [18] apply learning direction theory to analyze agents in an oligopolistic environment with restricted information and Golman and Page [8] study basins of attraction and equilibrium selection under different learning rules. Nakov and Nuño [19] use mechanisms similar to reinforcement learning to simulate learning of economic agents on stock markets and Lahkar and Seymour [13] apply reinforcement learning to show that agents in a population game revise mixed strategies. An overview of learning methods can be found in [3,6]. This paper analyzes a learning method inspired by reinforcement learning called Nash Q-learning [11].

Utilizing an agent-based simulation and the reinforcement learning methodology, we explore how adaptive agents without perfect information behave in Hotelling's location model [10] with quadratic consumer cost functions. Comparative analysis between the theoretical findings and the results of the agent-based simulation is provided. Additionally, we evaluate the reinforcement learning suitability for use in economic agent-based simulations and compare it to other learning methods.

Hotelling's location model is a microeconomic model presented by Harold Hotelling in 1929. The author found that two rational producers in the same market should make their products as similar as possible [10]. This phenomenon is called the Principle of Minimum Differentiation. Nevertheless, it has been shown that Hotelling's conclusions regarding minimum differentiation are invalid, and based on Hotelling's argumentation, "nothing can be said about the tendency of both sellers to agglomerate at the center of the market" [4, p. 1145]. Slightly modified versions of the location model with different consumer cost functions have been proposed, where the Principle of Maximum Differentiation [4] and the Principle of General Differentiation [5] can be justified. However, Hanaki et al. [9] analytically and numerically show that for $n, n \geqslant 2$, boundedly rational players following myopic best-reply strategy, the players spend most of the time around the center of Hotelling's street, which could re-establish Hotelling's Principle of Minimum Differentiation. Similarly, Matsumura et al. [16] show that minimum differentiation could be realized with evolutionary dynamics. Also, according to Bester et al. [1, p. 165], there are infinitely many mixed strategies in Hotelling's location game, in which "coordination failure invalidates the Principle of 'Maximum Differentiation' and firms may even locate at the same point".

The simulation contains two self-interested agents competing in a location model framework. The agents have no previous knowledge of the game or the opponent. The agents' main challenge is to communicate respective preferences through mutual interaction, learn about the game pay-offs, and try to find the Nash equilibrium strategy profile of the game. Every round of the simulation, agents have to choose what direction to move (location change) and what price to charge (price change). After taking actions, they receive information about their opponent's action and thus also about the current state of the game. They also receive positive or negative feedback based on how well they played in that particular round. The feedback is constructed in compliance with Hotelling's profit function. Agents can see theirs as well as their opponent's profit. The Nash Q-learning algorithm by Hu and Wellman [11] is used to guide our agents' adaptive behavior. Since Hotelling's location model contains convenient symmetries, agents learn not only from their experience but also from their opponent's experience. That is, both agents model their opponent as if they were the opponent.

The paper proceeds as follows. Section 1 provides details of Hotelling's location model. In Section 2, we theoretically discuss the learning methods. In Section 3, we describe technical details of the implementation, and Section 4 summarizes and interprets the important results of our simulation-based analysis. The penultimate Section 5 addresses several technical issues of our pioneering approach that might introduce open questions for future research. Finally, in Section 6, we conclude the paper with a summary of the crucial findings. The complete code is available on GitHub.

1. Hotelling's location model

The location model by Hotelling [10] introduces a strategic game among two producers. The game can be divided into two stages. In the second stage (short-run), producers compete in prices given to a fixed pair of locations $c_1, c_2 \in [a, b]$. In the first stage (long-run), producers compete in a location given that they adjust the Nash equilibrium in prices instantly. Then a combination of prices and locations $(p_l^*, p_r^*, c_l^*, c_r^*)$ is a pure strategy Nash equilibrium if p_l^*, p_r^* are the Nash equilibrium prices for locations c_l^*, c_r^* and locations c_l^*, c_r^* are the Nash equilibrium locations given prices p_l^*, p_r^*.

Customers are uniformly distributed on the $[a, b]$ interval and each customer demands exactly one product unit. Their demand is perfectly inelastic, meaning that they will demand exactly one product item irrespective of price. Transportation

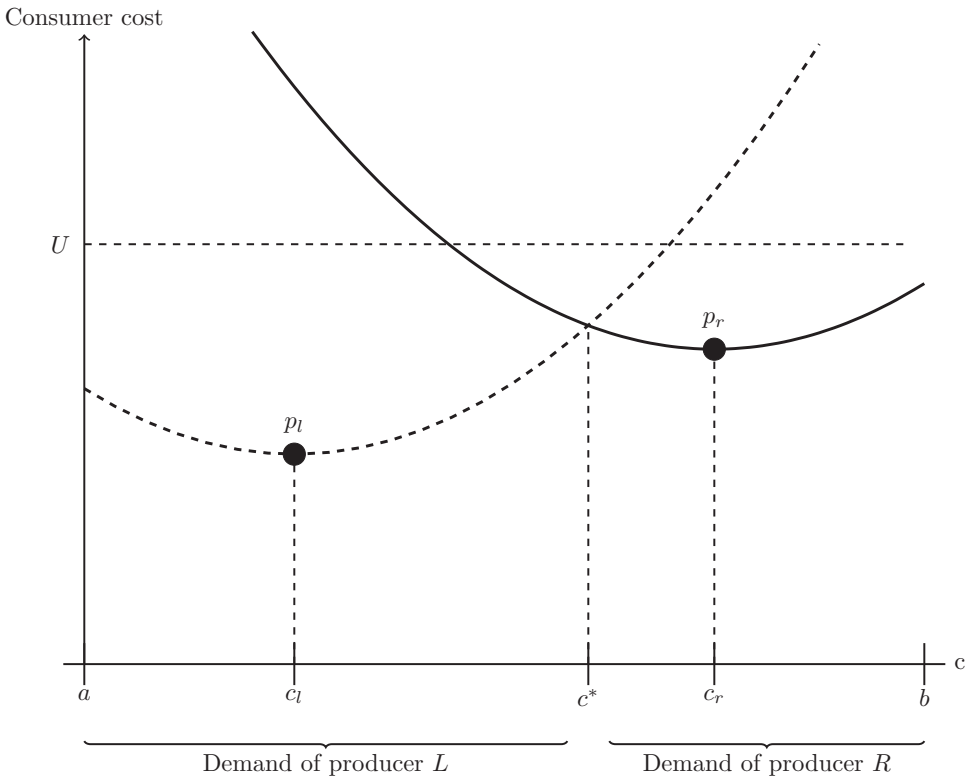


Fig. 1. Customer cost functions as functions of the location of customers. *Note:* The dashed function represents the cost for customers if they buy the leftmost product. The filled function represents cost for the rightmost product. The utility of each customer of buying a product is sufficiently high, so that their demand is perfectly inelastic. Cost functions are quadratic in distance.

cost is a function of distance from a customer's preferred characteristic of the product he or she would like to buy and an actual characteristic of the product he or she buys. On the $[a, b]$ interval, transportation cost is a function of distance between two points. If we define transportation cost as a function of distance $f(d)$, such that f grows with d, then it is possible to define the utility function of a customer. This principle is visualized in Fig. 1.

Definition 1.1. Let $i \in \{l, r\}$. Then let p_i be a price that producer i charges for his product and $f(d)$ is a transportation cost growing in d, and d_i is the Euclidean distance between customer c and producer i. If the utility U derived from owning the product is sufficiently large, then the utility of buying the product from producer i for a customer located at point $c \in [a, b]$ can be defined as

$$u_c(p_i, d_i) = U - p_i - f(d_i), \tag{1}$$

$$u_c(p_i, d_i) \geqslant 0\,. \tag{2}$$

In the original model [10], $f(d) = qd, q > 0$ so the transportation cost is a linear function of distance. d'Aspremont et al. [4] show that under linear transportation costs, there is no pure strategy Nash equilibrium in the model. For the Nash equilibrium in prices to exist in the original model, the producers have to be sufficiently far apart. If the producers are in locations with price equilibrium, there is an incentive to move closer to the opponent. But then they have to reach the distance zone, where there is no equilibrium in prices. However, if we allow $f(d) = qd^2$ as suggested by d'Aspremont et al. [4], then there is a unique pure strategy Nash equilibrium in both stages of the game and thus in the whole game. Since our agents try to find the Nash equilibrium of a game, a version of the location model where there is a known pure strategy Nash equilibrium is preferred. For this reason, we apply a version of location model with quadratic transportation costs suggested by d'Aspremont et al. [4], where price equilibrium exists for every location of the agents and a pure Nash equilibrium exists.

Based on locations and prices of the producers and on the utility function of customers, it is possible to derive demand for both producers. Without loss of generality, we can define A as the distance of the leftmost producer from point a and B as the distance of the rightmost producer from point b. The situation is illustrated in Fig. 2.

Since both products provide the same utility U, a customer c will choose the product j that satisfies $j = \arg\max_i u_c(p_i, d_i) = \arg\ \min_i p_i + f(d_i)$. The population of customers is divided into two parts, where the central customer

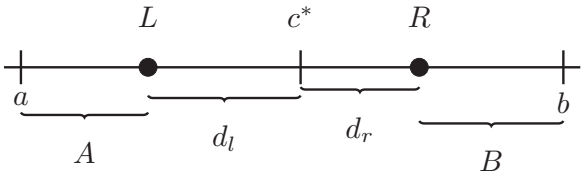


Fig. 2. Hotelling's market. *Note:* $[a, b]$ is the interval on which the customers are uniformly distributed, L and R depict the locations of the producers. c^* is the position of the central customer. A, B, d_l, and d_r are the respective distances.

c^* in between those groups shall be indifferent between both producers. This leads us to Eq. (3), where subscripts l and r denote the leftmost and rightmost producer

$$\begin{aligned} u_c(p_l, d_l) &= u_c(p_r, d_r), \\ p_l + q d_l^2 &= p_r + q d_r^2. \end{aligned} \tag{3}$$

As we can see from Fig. 2, for the central customer it has to hold that

$$A + d_l + d_r + B = |b - a|. \tag{4}$$

Once we combine Eq. 3 and Eq. 4, we end up with a linear system of equations. Solving this system for d_l and d_r leads to

$$\begin{aligned} d_l^* &= \frac{p_r - p_l}{2qz} + \frac{z}{2}, \\ d_r^* &= \frac{p_l - p_r}{2qz} + \frac{z}{2}, \end{aligned} \tag{5}$$

where $z = |b - a| - A - B$ and q is a positive constant. Then for the leftmost producer, the demand is

$$D_l(p_l, p_r, A, B) = \begin{cases} A + d_l^* & \text{if } A + d_l^* \in [0, |b-a|], \\ |b - a| & \text{if } A + d_l^* > |b-a|, \\ 0 & \text{if } A + d_l^* < 0. \end{cases} \tag{6}$$

Similarly, for the rightmost producer. Note, that for $A + B = |b - a|$ the d^* is undefined. In such case the producers show no differentiation and equilibrium prices are zero.

1.1. Price competition

To calculate a price Nash equilibrium, we just have to construct profit functions of both producers from the demand functions. Let Π_l denote the profit of the leftmost producer. Then

$$\Pi_l(p_l, p_r, A, B) = D_l(p_l, p_r, A, B) \cdot p_l, \tag{7}$$

and analogously for the rightmost producer. To arrive at a pure price Nash equilibrium given some locations, we first define best response functions

$$\begin{aligned} \mathrm{BR}_l(p_r) \in [0, \infty] : \frac{\partial \Pi_l}{\partial p_l} &= A + \frac{z}{2} + \frac{p_r - 2p_l}{2qz} = 0, \\ \mathrm{BR}_r(p_l) \in [0, \infty] : \frac{\partial \Pi_l}{\partial p_l} &= B + \frac{z}{2} + \frac{p_l - 2p_r}{2qz} = 0. \end{aligned} \tag{8}$$

Along with Eq. 8, we have $\frac{\partial^2 \Pi_l}{\partial p_l} = \frac{\partial^2 \Pi_r}{\partial p_r} = \frac{-2}{2qz} < 0$ ensuring that profit functions are actually maximized. Solving Eq. 8 leads to the best response price $\mathrm{BR}_l(p_r) = Aqz + \frac{qz^2 + p_r}{2}$. Repeating for the rightmost player, we get $\mathrm{BR}_r(p_l) = Bqz + \frac{qz^2 + p_l}{2}$. If a pure Nash equilibrium in prices exists, it is a vector of prices (p_l^*, p_r^*), such that $p_l^* = \mathrm{BR}_l\Big(\mathrm{BR}_r(p_l^*)\Big)$ and similarly for p_r^*. The Nash equilibrium in this case is unique. The equilibrium prices are

$$\begin{aligned} p_l^* &= qz\Big(\tfrac{A-B}{3} + |b - a|\Big), \\ p_r^* &= qz\Big(\tfrac{B-A}{3} + |b - a|\Big). \end{aligned} \tag{9}$$

Notice that when producers are located in the same place, the equilibrium price turns out to be 0, there is no product differentiation, and producers only compete in price. As the producers move further away from each other, they create local monopolies and increase their price above their marginal cost.

1.2. Location competition

To establish the second-stage location equilibrium, we substitute the Nash equilibrium prices to profit functions and differentiate the profit functions with respect to location. We obtain

$$\begin{aligned} \frac{\partial \Pi_l}{\partial A} &< 0, \\ \frac{\partial \Pi_r}{\partial B} &< 0. \end{aligned} \tag{10}$$

This implies that in the long run the producers have an incentive to get further from each other for any initial location. The long-term location Nash equilibrium is the vector (a, b) with equilibrium prices $p_r^* = p_l^* = q|b-a|^2$, since $A = B = 0$. This also means that the long-term equilibrium profit is

$$\Pi_{\text{NE}} = \frac{q|b-a|^3}{2}. \tag{11}$$

These theoretical results originally established by d'Aspremont et al. [4] are used to evaluate the learning results of agents in our simulation.

2. Learning methods

2.1. Reinforcement learning

Many times, humans and animals alike have no explicit teacher who would show them how to act in an unknown environment. Nevertheless, through experimenting with available actions and a recognition of how the actions influence the world that humans and animals perceive, they are able to form conclusions about cause and effect, consequences of actions, and about what to do in order to achieve goals [23]. According to the authors, agents in a reinforcement learning setting learn how to choose reward-maximizing actions in given situations by means of trial-and-error. A given action does not only influence the current reward but also potentially all future rewards. Various methods can be used to solve a reinforcement learning problem. For example, Q-learning [25] and SARSA [23] algorithms can be used to optimize a single agent's behavior in a stochastic environment.

A reinforcement learning agent does not necessarily have any information about the environment, but has to be able to balance an exploration of the environment with choosing an optimal strategy based on current environment estimates and in the long run, he or she is able to find an optimal strategy. A reinforcement learning problem can be expressed with a Markov decision process, a stochastic process in which the agent has certain *actions* available for each state of the environment. Each action has certain transition probability to other states of the environment. By choosing actions, the agent moves from state to state. Each transition emits a *reward*. A rational agent maximizes the long-term reward that comes from individual transitions, that is, he or she maximizes the discounted infinite sum of future rewards.

If we place more agents inside a non-changing environment, the environment as perceived by the agents is changing if the agents are allowed to evolve. The reason is that once one agent adapts to the other agent's strategy, the other agent should adapt as well. We can observe these adaptation cycles in nature in the predator-prey relation: prey adapts to predators and predators adapt to prey. Such environment can be modeled with *stochastic games*.

Definition 2.1 (Stochastic game)**.** A stochastic game with $n \in \mathbb{N}$ players is defined by a tuple $(\mathcal{S}, \mathcal{A}_s^1, \dots, \mathcal{A}_s^n, r^1, \dots, r^n, p)$ where

- $\mathcal{S}$ is the state space,
- $\mathcal{A}_s^i$ is the set of actions available to player $i \in 1, \dots, n$ in state $s \in \mathcal{S}$,
- $r^i : \mathcal{S} \times \{\times_{i=1}^n \mathcal{A}_s^i\} \to \mathbb{R}$ is a payoff function for player i,
- $p : \mathcal{S} \times \{\times_{i=1}^n \mathcal{A}_s^i\} \to \Delta(\mathcal{S})$ is the transition probability map, where $\Delta(\mathcal{S})$ is the set of probability distributions over $\mathcal{S}$.

This definition is according to Hu and Wellman [11] and includes both the transition dynamics as well as rewards depending on the joint actions of the decision makers and on the state they are in. Let $p(\cdot|s, a^1, \dots, a^n) \in \Delta(\mathcal{S})$. If we restrict ourselves to stochastic games with finite state and action sets, then

$$p(\cdot|s, a^1, \dots, a^n) : \mathcal{S} \to [0, 1], \text{ such that } \sum_{s' \in \mathcal{S}} p(s'|s, a^1, \dots, a^n) = 1, \tag{12}$$

where $s \in \mathcal{S}$ and $a^i \in \mathcal{A}_s^i$ for all i. Let π^i be a policy or strategy of player i, λ_i be the discount factor of player i and $G_t^i = \sum_{k=0}^{\infty} \lambda_i^k R_{t+k+1}^i$. Then the decision maker i should try to maximize a value function

$$v_\pi^i(s) = \mathbb{E}_\pi[G_t^i | S_t = s], \tag{13}$$

where $\pi = \pi^i, \ldots, \pi^n$ is a *strategy profile*.[1] The optimal strategy of one player is conditioned on the strategies of all the other players. Such a strategy is called the *best response*. We can define a Q-value function associated with the value function v. A Nash equilibrium q-value function can be defined as

Definition 2.2 (Nash q-value function)**.** Let $\pi = \pi^1 \ldots \pi^n$ be a Nash equilibrium strategy profile of a stochastic game, $a \in \times_{i=1}^{n} \mathcal{A}_s^i$ and $s \in \mathcal{S}$. Then a q-value function for (s, a) given π for player i is

$$q_\pi^i(a, s) = \sum_{s'} p(s'|s, a)[r^i(s, a, s') + \lambda v_\pi^i(s')], \tag{14}$$

where $v_\pi^i(s')$ is the value function in state s' [11].

If $\pi = \pi*$ is a Nash equilibrium profile, then the q-value function can be written as

$$q_{\pi*}^i(a, s) = \sum_{s'} p(s'|s, a)[r^i(s, a, s') + \lambda v_{\pi*}^i(s')] \tag{15}$$

$$= \sum_{s'} p(s'|s, a)[r^i(s, a, s') + \lambda \underset{a}{\text{Nash}}\{q_{\pi*}(s', a)\}], \tag{16}$$

where the Nash operator in (16) calculates the Nash equilibrium action profile based on the values of $q_{\pi*}^i(s', a)\}, i \in \{1, \ldots, n\}, a \in \times_{i=1}^{n} \mathcal{A}_{s'}^i$ and returns the expected future discounted Nash equilibrium pay-off for player i. There is an essential difference between the Nash equilibrium of a stochastic game and the Nash equilibrium calculated in the Nash operator. The former refers to policy functions. The latter refers to actions taken in a particular state and is calculated from a matrix game consisting of q-values for a given state and available actions. In the paper by Hu and Wellman [11], the latter type of game is called a *stage game*.

A Nash equilibrium of a game is a strategy profile from which nobody wants to deviate. In other words, all players play their best response strategy to other players' strategies. Thus, all players maximize their Q-value functions with respect to other players' strategies. Notice that there can be more Nash equilibria profiles supporting different value functions.

Among research papers extending ideas from reinforcement learning to multi-agent domain are those introducing the methods of Friend-or-Foe Q-learning by [14] and Nash Q-learning by [11]. Both methods proposed by given papers attempt to find a Nash equilibrium of the system and can be applied to general-sum stochastic games, but require relatively strong conditions to be successful. We use the latter method for our simulation.

2.2. Nash Q-learning

This section describes the Nash Q-learning algorithm. Nash Q-learning can be utilized to solve a reinforcement learning problem, where there are multiple agents in the environment. In contrast to single-agent methods, the Nash Q-learning agent takes other agent's goals into account. Therefore, Nash Q-learning agents provide a modeling tool that is more realistic than single-agent methods, where the other agent's utility functions are not estimated. The goal of agents in a multi-agent environment in general terms is to maximize their value function with respect to other players' strategies. One possible goal of agents in a stochastic game is to play a Nash equilibrium profile, because in such a profile, everyone's value function is maximized.

The goal of Nash Q-learning agents is to learn a Nash equilibrium q-value function q_*^i and a Nash equilibrium strategy profile $\pi*$ associated with the q-value function. The main idea behind Nash Q-learning is that each player estimates the q-values of all other players as well as of theirs. Then each player estimates other players' q-functions and is able to calculate Nash equilibrium profiles in a matrix game. The update rule is defined as follows.

Definition 2.3 (Nash Q-update rule)**.** The Nash Q-update rule of a q-value function estimate for player i in time $t + 1$ is

$$\begin{aligned} \hat{q}_{t+1}^i(s, a) = {} & (1 - \alpha_t)\ \hat{q}_t^i(s, a) \\ & + \alpha_t[r_t^i(s, a, s') + \lambda \cdot \underset{a'}{\text{Nash}}\ \hat{q}_t^i(s', a')], \end{aligned} \tag{17}$$

where the Nash operator at time t calculates a Nash equilibrium pay-off from an estimated stage game $\hat{Q}_t(S_t) = \{q_t^k(s, a) \,|\, a \in \times\{\times_{i=1}^{n} \mathcal{A}_s^i\}\}$ and $\alpha_t = 0$ if $(S_t, A_t) \neq (s, t)$ [11].

According to Hu and Wellman [11], the estimator from definition 2.3 converges in probability to q^i under the following conditions:

i. Every state and action of the stochastic game are visited infinitely often.
ii. The learning rate α_t satisfies the following conditions:

[1] A player's policy is a mapping from game states to that player's actions. A strategy profile is a permutation of the actions of all players. A strategy profile can be seen as a cartesian product of the policies of all players.

Table 1
A 2-player stage game with a global optimal and a saddle point.

	a_1^2	a_2^2
a_1^1	10,10	0,6
a_1^2	6,0	5,5

Note: The global optimal point is in the top left corner and the saddle point is in the bottom right corner.

(a) $0 \leqslant \alpha_t(s,a) < 1,\ \sum_{t=0}^{\infty} \alpha_t(s,a) = \infty, \sum_{t=0}^{\infty} \alpha_t^2(s,a) < \infty,$
(b) $\alpha_t(s,a) = 0$, if $(s,a) \neq (S_t, A_t)$.

iii. Every stage game $\hat{Q}_t(s)$ in the learning process has a global optimal point and agents' pay-offs in these equilibria are used to update their $\hat{q}$-functions, or every stage game $\hat{Q}_t(s)$ in the learning process has a saddle point and agents' pay-offs in these equilibria are used to update their $\hat{q}$-functions.

The first two conditions are identical to the convergence conditions for single-agent Q-learning. The third condition requires that all stage games arising during learning must have a global optimal point, or alternatively a saddle point. A global optimal point is an action profile such that all players receive their highest pay-offs at that profile, which consequently also makes it a Nash equilibrium. A saddle point is a Nash equilibrium action profile such that each player receives higher pay-off if at least one of the other players deviates [11]. The third condition further requires that players always update their $\hat{q}$-functions according to pay-offs in global optimal point or alternatively always in a saddle point. An example of a stage game with a global optimal point and a saddle point is displayed in Table 1.

The third condition would generally be very difficult to satisfy. One is usually not able to guarantee that all stage games emerging during learning will satisfy it. However, Hu and Wellman [11] suggested that there may be some potential to relax the convergence conditions. They tested the Nash Q-learning algorithm on a set of gridworld games. In the game that satisfied the third condition at the stochastic game Nash equilibrium, there was a convergence of the learned $\hat{q}$-values to q 100% of time despite the fact that during learning, the third condition was not met. Detailed convergence results and comparisons of learning performance can be found in Hu and Wellman [11, p. 1060, 1061].

3. Implementation

The original Hotelling's model is a differential game, where the seller's location and price are continuous variables. For simulation purposes, we discretized the price and location space. Thus, the game played by our agents is no longer smooth, and there can be multiple equilibria. Moreover, the original game is played only once. Since learning happens iteratively, we added a time dimension to our simulation. The agents cannot freely select their location in a single step but have to travel for several steps to reach their destinations. This makes our setup more similar to a repeated game, where the agents have to optimize towards the future. It also brings the whole setup closer to real-world situations.

The previous sections show in general how agents can potentially learn in a multi-agent environment. In this section, we present our simulation implementation details. The simulation is discrete in every parameter. Location and price sets are both finite. Time steps are discrete. The simulation runs 30,000 time steps; that is, the agents interact with each other and with their environment 30,000 times. We repeat the simulation 100 times for each combination of parameters specified below to support statistical validity. The behavior of agents almost does not change after 30,000 time steps, suggesting the numerically established asymptotic behavior of the agents.

All agents have a decreasing learning rate. A learning rate for an agent $\alpha_t(s,a)$ is defined as

$$\alpha_t(s,a) = \begin{cases} \frac{1}{n_t(s,a)} & \text{if } n_t(s,a) > 0, \\ 1 & \text{if } n_t(s,a) = 0, \end{cases} \tag{18}$$

where $n_t(s,a)$ is a number of times the state-action pair (s,a) was visited. Learning rate satisfies the convergence condition specified in Section 2.2. Exploration rate is set to 0.2. Exploration rate is greater than zero and thus there is a non-zero probability of exploration in every state. An agent explores randomly if a random number generated from (0,1) interval is smaller than the exploration rate. Otherwise, the agent chooses the first Nash equilibrium strategy calculated from his stage-game estimates. The discount rate λ can differ from agent to agent. Based on preliminary experiments,[2] we choose the combinations displayed in Table 2.

In the first case, the agents are perfectly impatient and only consider current rewards. In the third case, one of the agents is perfectly impatient while the other one discounts his future rewards by factor 0.8. The remaining are various combinations of impatient, patient, and semi-patient agents.

[2] Available upon request from the authors.

Table 2
Used combinations of λ.

λ_1	λ_2
0	0
0	0.4
0	0.8
0.4	0.4
0.4	0.8
0.8	0.8

Note: Simulation of reversed permutations is not necessary, because the game is symmetric in profits.

We choose the location set to be $L = \{0, 1, \ldots, 6\}$, that is, the middle of the 'street' is at number 3. Agents can set the price of their product to one of the price levels from $P = \{0, 1, 2, 3, 4\}$. Both agents can choose any price from P in every price game. Agents do not choose locations directly but choose moves instead. Available moves are $M = \{-1, 0, 1\}$ so an agent can move one location to the left, one location to the right, or stay in his or her original position. Agents can stand on the same location but cannot escape the street. In other words, action set available on the edges of the street does not contain the action that would move the agents outside the street. The location set is chosen to be reasonably large to capture potentially interesting behavioral patterns, yet not too large because at every time step the Nash equilibria are calculated and a too-large location set would cause unnecessarily long execution of the simulation. The q parameter of the transportation function described in Section 1 is set to 1.

The whole simulation is written in *Python 3.6.6*. Agents use the *nashpy 0.0.17* library to calculate the Nash equilibria of stage games. The method used to calculate Nash equilibria is Lemke-Howson. We have already discussed that Stage games in general do not have to satisfy the Nash Q-learning conditions. Equilibrium selection is an open problem in Game theory. Thus, to make the whole simulation tractable, our agents always choose the first Nash equilibrium strategy provided by the Lemke-Howson method, similarly to Hu and Wellman [11]. For the generation of random numbers we use the *random* module from *numpy 1.15.4* library. The complete code is available on GitHub at the following address: github.com/janvainer/agent_based_hotelling [created 27 November 2018].

4. Simulation results

4.1. Asymptotic behavior

First, we summarize the asymptotic behavior of key metrics for our agents. We observe the development of prices, locations, and profits in our simulation. From the location we also calculate the distance between the agents at every time step. The development of all parameters is captured for 30,000 time steps. To soften the raw data we use rolling averages of various lengths according to given situations. Let us now only concentrate on profits. We take profits from all 100 simulation runs for each discount rate combination. A moving average transformation is applied to each simulation run. Next, for each discount rate at every time step, an arithmetic average across all simulation runs at the given time step is taken. Resulting time series are depicted in Fig. 3.

We can observe how the average profit grows with time. However, the growth rate decreases and average profits seem to be relatively stabilized after 25,000 time steps signaling that the learned polices of the agents do not significantly change any more.

Long-term planning brings higher profit. High discount rates mean that agents optimize more heavily towards more distant rewards. Thus, we would expect the agents with high discount rates to be better in long-term planning and thus earn better profits than short-sighted agents. This is confirmed for the discount rate combinations where one agent has a higher discount rate than the other one. To observe this result, we compare Fig. 4 and Fig. 5. If the discount rates are identical, the profits are identical as well, and vice versa.

Interestingly, the agents with highest discount rates $\lambda_1 = \lambda_2 = 0.8$ reach the lowest average profits as we can observe in Fig. 3. This implies that the Nash equilibrium profit in our simulation where agents consider long-term rewards could be different from the short-term equilibrium profits, where agents only take into account recent profits. To be able to explain this phenomenon, we need to zoom into the simulation data and analyze if there are some pronounced patterns of behavior and whether those patterns differ for high and low discount rate combinations.

4.2. Location and price dynamics

Now we plot a detailed segment of one of the simulation runs for identical discount rates in Fig. 6. Surprisingly, there seems to be tendency to oscillate in location. The observed behavior clearly differs from Hotelling's location model with

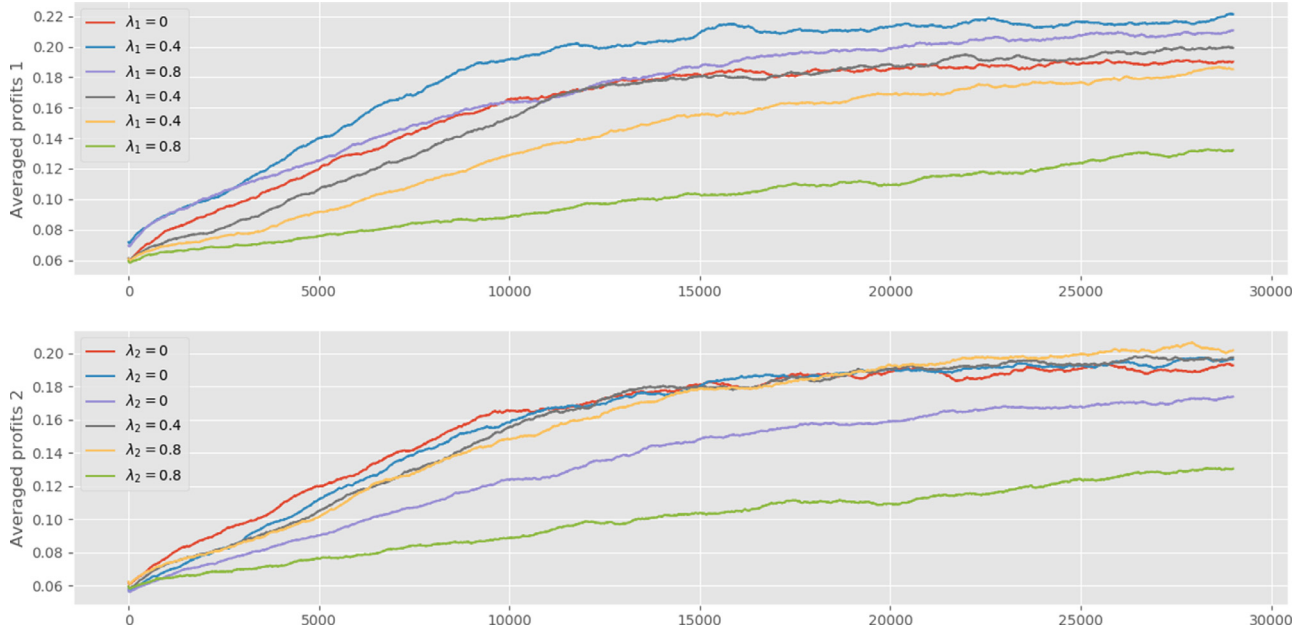


Fig. 3. Average profits. *Note:* Moving average profits for each discount rate combination is averaged across all 100 simulations at each time step. The upper sub-plot shows average profits of first agents and bottom sub-plot shows average profits of second agents. For instance, the combination of red lines shows the development of profits in the simulation where both agents have a discount rate set to 0. Length of rolling average is $n = 1,000$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

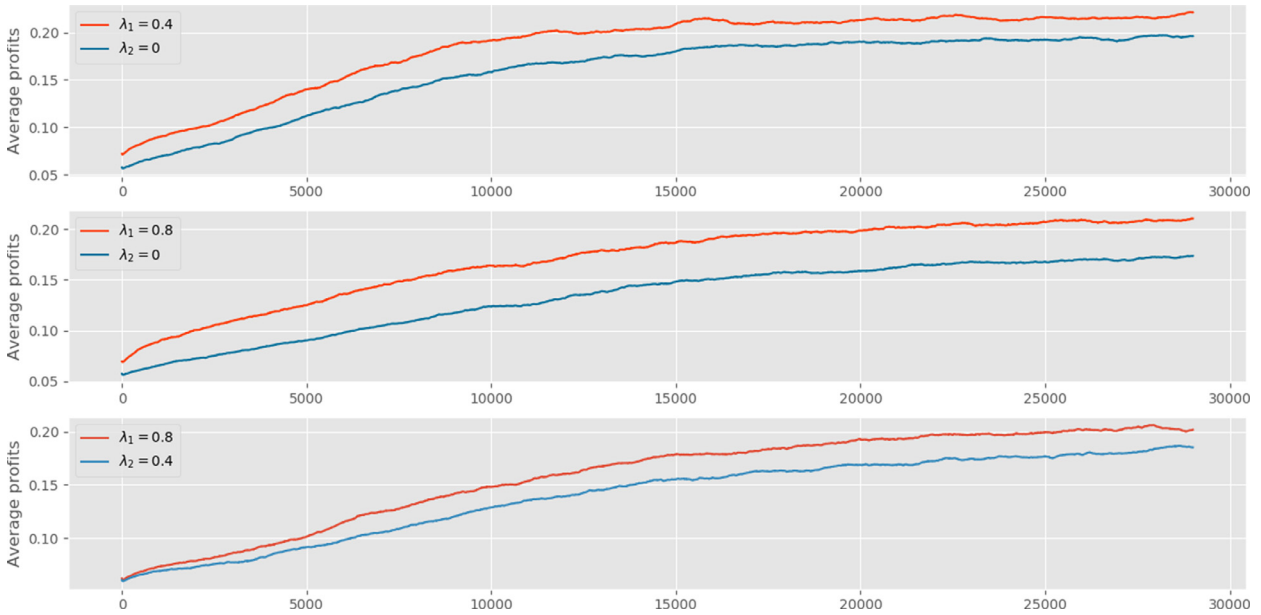


Fig. 4. Average profits of agents with unequal discount rates. *Note:* Length of the rolling average is $n = 1,000$.

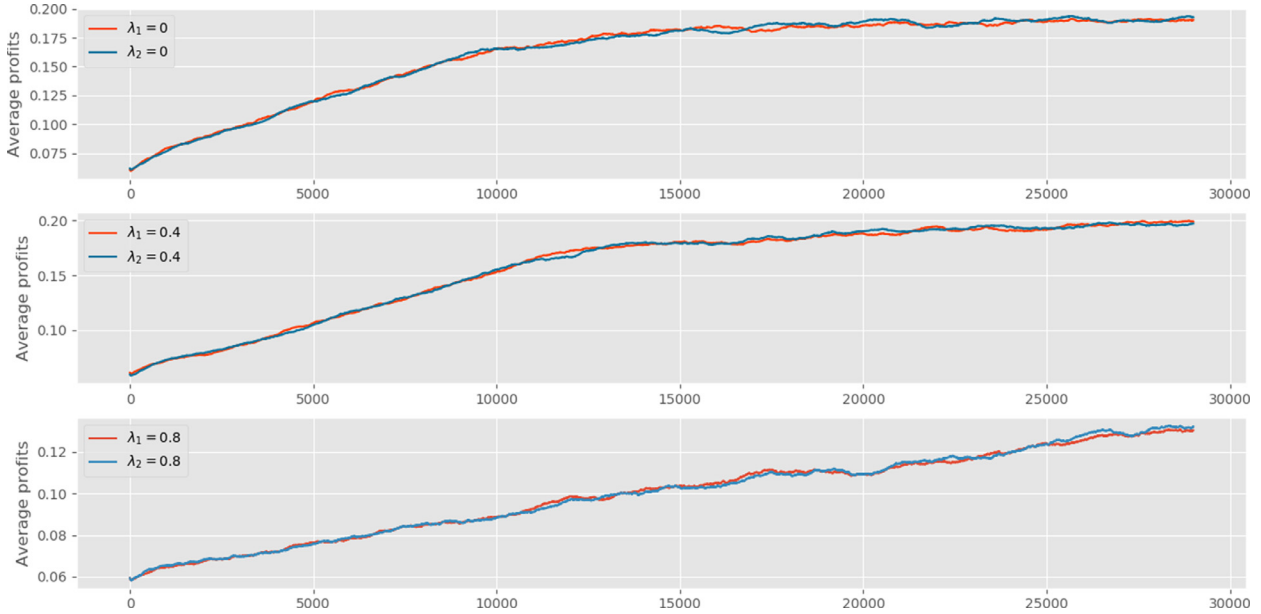


Fig. 5. Average profits of agents with equal discount rates. *Note:* Length of the rolling average is $n = 1,000$.

quadratic consumer costs, where the agents stay on the opposite edges of the street and do not periodically meet in the center.

Notice that in some cases agents switch their positions but in other cases they just meet and then get back to their previous locations. We can observe this pattern in Fig. 7, where agents meet at the beginning and at the end of the segment without changing sides. This can be explained by occasional random exploration that pushes agents from each other to a state where their strategy already is such that they get further away. Regarding profit dynamics, changing or not changing sides does not seem to make difference. In both cases, the profits decrease and increase again once the agents reach opposite edges of the street. However, the profit level seems to be higher for the agent occupying the upper part of the street. For example, between time steps 25,200 and 25,250 in Fig. 6, the red profit line is mostly above the blue line. This situation gets reversed once the agents switch their sides of the street. The difference in profit levels between the upper and lower part of the street can be explained by the discretization of the location and price space.

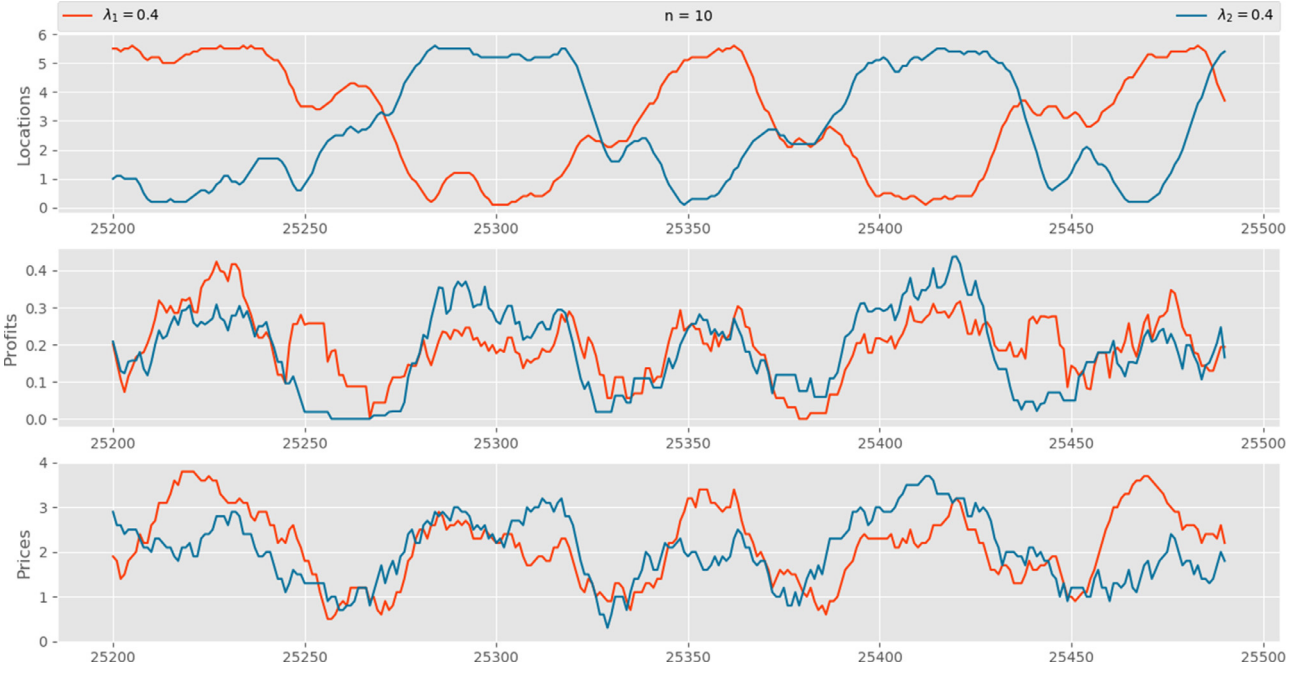


Fig. 6. A detailed view of a segment of a simulation run. *Note:* A random data segment from a simulation run of agents with identical discount factors $\lambda_1 = \lambda_2 = 0.4$. The length of the rolling average is $n = 10$. A range of data between 25,200 and 25,490 time steps is displayed.

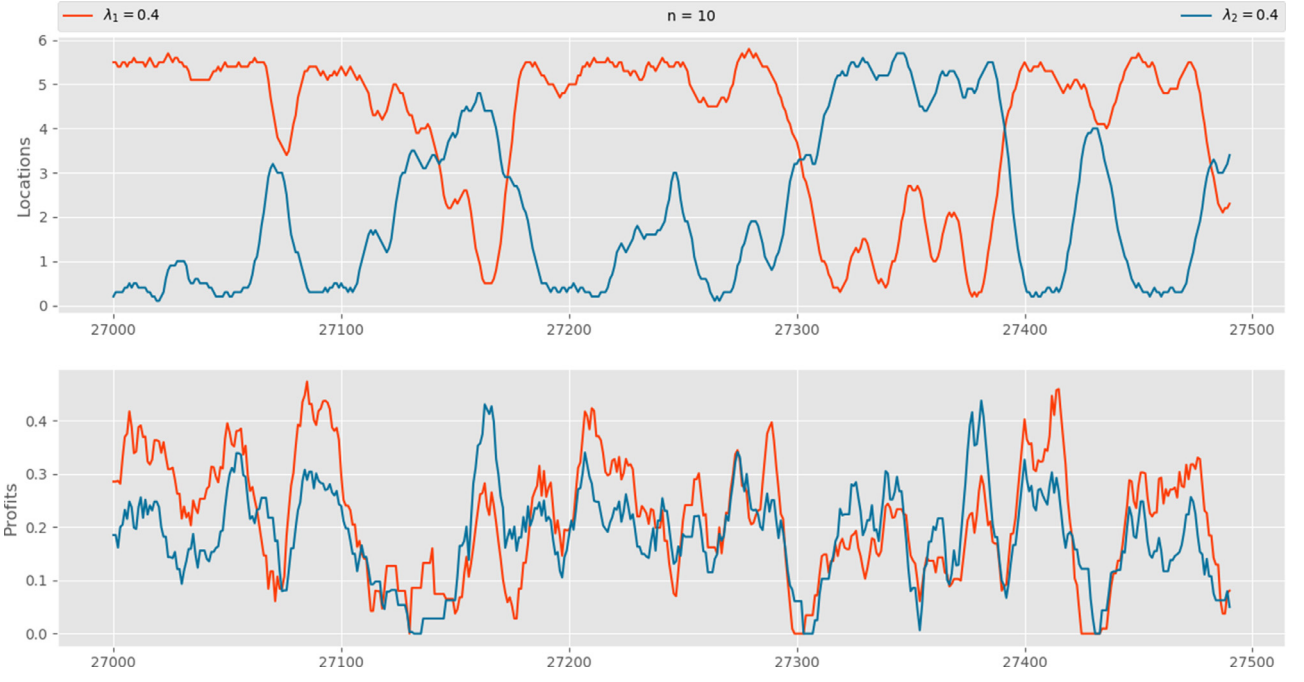


Fig. 7. Agents can meet without switching sides. *Note:* A data segment from a simulation run of agents with identical discount factors $\lambda_1 = \lambda_2 = 0.4$, where agents meet twice but do not change their sides. The length of the rolling average is $n = 10$. A range of data between 27,000 and 27,490 time steps is displayed.

Discretization can lead to the creation of multiple pure Nash equilibria. This holds also for situations where there was only one pure equilibrium in the continuous case. Some of the equilibria may be asymmetric in pay-off. If our discretization creates such equilibrium and this equilibrium gets calculated first in the Nash equilibrium calculation, then the agents learn to follow this asymmetric equilibrium.

Payoff asymmetry can motivate agents to switch their positions. The agent on the worse side of the street can have a long-run incentive to leave his or her side. he or she thus challenges his or her opponent and tries to steal some of his or her opponent's demand. The only reasonable response of the opponent to such behavior would be to accept the challenge. Otherwise, the agent who steals the demand can get even closer to his or her opponent and enclose his or her in the edge of the street and cut his or her opponent's profit completely. Thus, both agents meet somewhere in the middle of the street and their profits are close to zero, because they only compete in price and have no local monopoly anymore. The challenger wants to switch sides, because he or she can get better profit on the other side of the street. The opponent knows that the challenger has large enough incentive to continue competing, which would result in zero profits for both of them in foreseeable future and accepts the exchange. Later, the roles are switched and the whole cycle repeats.

There seems to be no profit of meeting in the middle and not switching for the agent who initialized the switch and there is no reasonable way that the other agent could stop the challenger from switching sides. We conclude that these situations are caused by random exploration at the time of the meeting.

Long-sighted agents have shorter site exchange periods. This happens because they put large weight on future rewards. Thus, the agent in the worse position would have immediate incentive to switch, because he or she would know how little he or she earns if he or she does not switch, and how much he or she would earn if he or she managed to switch positions with his or her opponent.

We can actually observe a similar pattern in Fig. 8. For the discount rate combination $\lambda_1 = \lambda_2 = 0.8$, the exchange period is much shorter than in other cases and the agents do not even reach the edges of the street. It starts if an agent gets into an unfavorable location. Then he or she evaluates how bad this location is in the long run and if it is sufficiently bad compared to an alternative, he or she decides to make the switch. But instead of increased profits for both agents, frequent exchange results in decreased profits. Lower profits of the agents with $\lambda_1 = \lambda_2 = 0.8$ are reported already in Fig. 3 and we now can find the reason in Fig. 8 in the last plot.

Two long-sighted agents mostly compete in price and not location. The agents almost do not leave each other and stay in very close locations, which reduces their local monopoly and forces them to compete mostly in price.

Long-sighted agents exploit short-sighted agents. If one of the agents has a higher discount rate, then he or she decides to leave the inconvenient side quickly, while the other agent with lower discount rate does not have as much incentive to switch. This is because the other agent is only interested in short-run profits and switching sides is not as attractive as it brings almost no income in the short-run. Thus, the long-sighted agent spends more time on the more profitable side of the street and as a result, he or she earns more profit on average. However, our interpretation could be slightly problematic for the case where $\lambda_1 = \lambda_2 = 0$. Those agents do not care about future rewards and only take immediate profit into account. Nevertheless, we still observe periodical switching of sides even in their case. Actually, in all our simulations we observe the location-switching pattern, however, the period generally differs and in many cases it is relatively long.

Side switching can be explained by random exploration. The interesting question emerging is as follows. If the agent in an unfavorable position has an incentive to switch at any point in time, why would he or she wait? A plausible explanation is the exploration factor. Possibly, the agents would stay on their side of the street like in Hotelling's location model. What forces them to exchange locations is the random exploration. They can be pushed by random shocks close enough to their opponent to a state where their opponent has to react and the switching happens. On the other hand, unlike in the previous cases, in the case of $\lambda_1 = \lambda_2 = 0.8$ we clearly observe that the agents almost never stay on the edges of the street and rather rapidly switch positions all the time.

We conclude that the fluctuations are likely to be caused by a combination of both factors we mention above. For most of the discount rate combinations it is not worth it to switch positions due to low profits during the switch and subjective low value of future profits. But once one of the agents cuts the distance between the two agents, the other one has to react and they switch positions and get back to street edges as if they were in Hotelling's equilibrium. They stay there until one of them again is pushed by random exploration closer to the opponent.

4.3. Similarity to the Hotelling's original model

The fact that agents with low discount rates seem to resemble the original equilibrium can be technically explained. For zero discount rates the agents only take immediate profits into account. Thus, the location agents only compare profits immediately available after their action, which is very similar to how the Nash equilibrium in Hotelling's model is calculated. Once price equilibrium in Hotelling's model is calculated for each location, the decision where to go in each location depends on infinitesimal changes in the profit caused by moving in one or the other direction. For the quadratic cost for consumers, the profit change is positive if the agents move further apart from each other, which causes the Nash equilibrium to be the edges of the street. In our case with low discount rates, agents put profits from moving to the left, right, or nowhere to a game matrix and calculate the Nash equilibrium of a simple matrix game in the given location. This is in principle very similar to the original model, because the profit values only contain immediate profit and not discounted future profits.

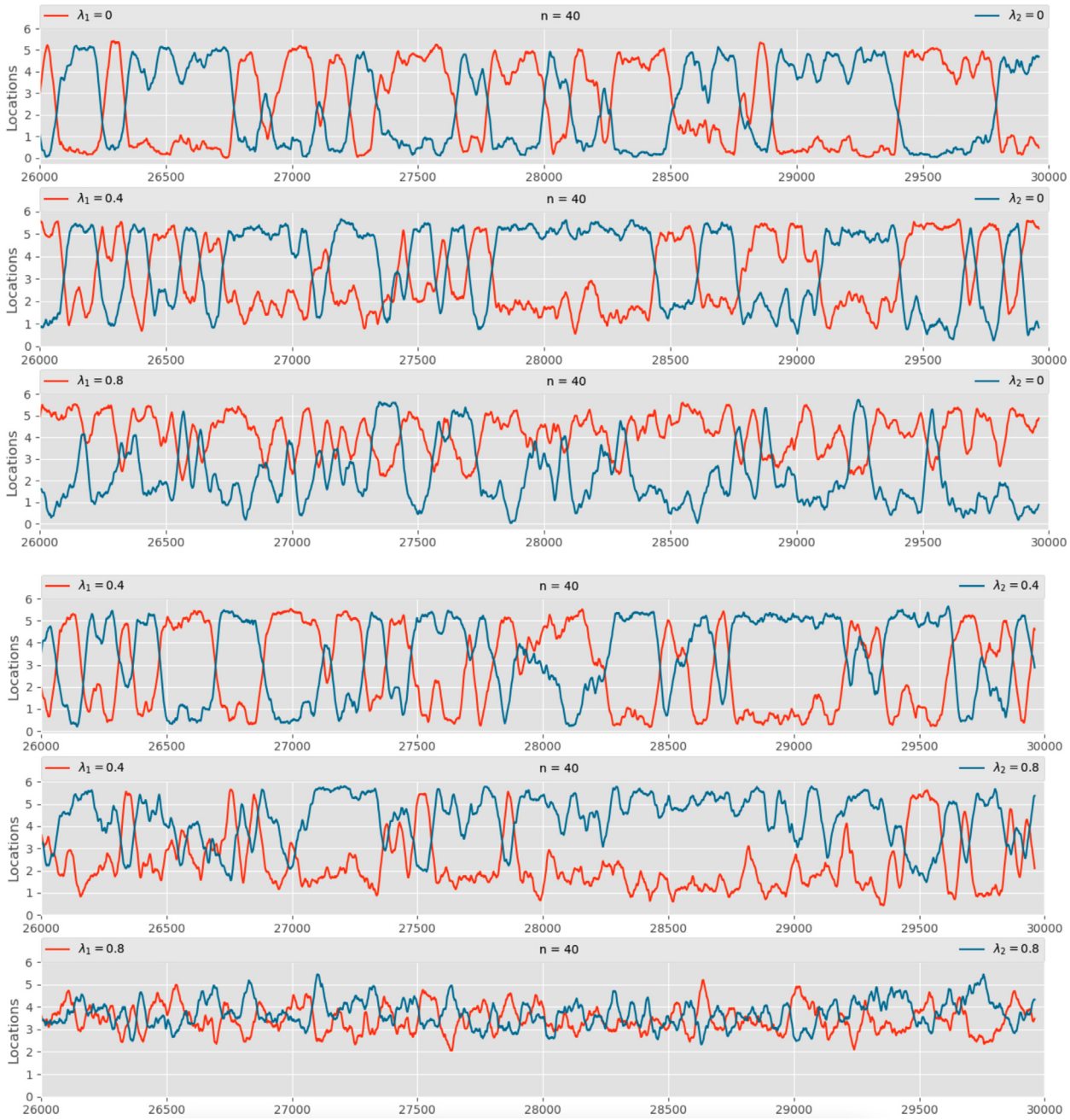


Fig. 8. Location dynamics of all discount rate combinations. *Note:* Data segments from simulation runs of all discount rate combinations. The length of the rolling average is $n = 40$. A range of data between 26,000 and 29,960 time steps is displayed.

Thus, the matrix game should have the same values that we work with in the original model and should therefore provide similar results.

In the case of $\lambda_1 = \lambda_2 = 0.8$, the agents seem to realize the long-term value of being in a more favorable position and a switch is an acceptable price to pay in comparison with potential profit on the other side of the street. Unfortunately for them, both agents want to be on the favorable side and they never settle and keep switching instead, which results in lower profits compared to other discount rate combinations. On the other hand, if the agent with $\lambda = 0.8$ is combined with an agent with a lower discount rate, he or she is able to keep herself on the more favorable side of the street most of the time. The other agent does not have an incentive to switch and if he or she does, it is just due to random exploration. Agents switch, but the long-sighted agent tries to switch back right after. We can see that on the second to last plot in Fig. 8. The blue agent is rarely on the lower part of the street and if he or she happens to appear there, he or she goes back almost

immediately. The same holds for the third plot from the top, where the red agent is mostly on the upper side of the street. The discount rate $\lambda = 0.4$ does not seem to value the future profits enough and those agents basically behave like the agents with zero discount rate.

The case where $\lambda_1 = \lambda_2 = 0.8$ approximately re-establishes the Nash equilibrium in the center of the street described by Hotelling in his original paper [10] with linear consumer costs, because the agents on average spend their time in the center and have very low profits. This behavior seems similar to an aggressive market strategy, when a company tries to eliminate his or her competition from the market by making similar products and leading a price war until either one of them goes bankrupt or both competitors exhaust themselves and start differentiating their products again. Because our agents have no production costs, they can never exhaust themselves. Thus, they locate themselves near each other at all times and compete mainly in price, which yields low profits.

4.4. Alternative setups and comparison

As a sensitivity check, we run two additional reference simulations with simpler agents. In the first alternative setup, we consider two random agents. Each agent selects their actions in both location and price domain at random. In the second alternative setup, we use two Q-learning agents. Compared to the Nash Q-learning agents, the Q-learning agents use the standard *max* operator instead of *Nash* operator for their *q-value* update rule (see Definition Eq. 2.3). The Q-learning agent's q-value does not explicitly take his opponent's actions and utility function into account. Therefore, the Q-learning agents behave as if their opponent was a static part of their environment. Such a simplification may not accurately model real economic actors because the opponent is learning and his strategy is changing over time. Otherwise, we employ the same simulation settings used for the Nash Q-learning simulations (Fig. 3). Results obtained from our alternative setups can be found in Appendix A.

The random agents' average profits in Fig. A.9 start at a value of 0.18 and remain constant throughout the simulation. This behavior is expected as the random agents do not change their behavior. Average profits of the Q-learning agents in Fig. A.10 start at approx. 0.2 and increase by 0.1 by the end of the simulation. Our Nash Q-learning agents start at average profits around 0.06 and climb up to values between 0.08 and 0.22 depending on the discount rate factor (see Fig. 3).

The profits of the Nash Q-learning agents are lower compared to those attained by the Q-learning agents. This can be explained by the fact that the Nash Q-learning agents try to find a stable strategy profile (i.e., nobody has a reason to change strategy) while the Q-learning agents simply maximize their profits. The main difference is that the Nash Q-learning agents take advantage of the knowledge of their opponent's estimated utility function, which may lead to higher profits. However, since both agents can try to exploit such knowledge simultaneously, their behavior may lead to low-profit equilibria similar to the situation in the *Prisoners dilema*.

4.5. Summary

We now summarize our discoveries. Instead of finding some location combination and settling there, our agents tend to periodically switch positions. Except for the $\lambda_1 = \lambda_2 = 0.8$ discount rate combination, the agents spend most of the time in opposite edges of the street. We conclude that switching in those cases is caused by random exploration that triggers a chain of events leading to a re-established balance, where agents again settle on edges of the street. In the case where $\lambda_1 = \lambda_2 = 0.8$, the agents stay near each other and fight for the more convenient side of the street indefinitely. This can be explained by both agents realizing the long-term value of the more convenient side of the street.

5. Technical issues

We take Hotelling's location model [10] and cut the continuous parameters from the model into discrete pieces in order to be able to run a discrete simulation. This is technically straightforward to accomplish. However, it brings the following issues:

i. On one hand, there is surely a Nash equilibrium in a finite game [20], on the other hand, this equilibrium may be non-unique and depends on the way we cut the parameters of the differential game.
ii. Due to non-uniqueness of the Nash equilibrium, our agents have to face the choice between available Nash equilibria.
iii. Agents may choose the mixed strategy Nash equilibrium if there no pure strategy equilibria, which is a situation that might be difficult to interpret economically.

Our results might depend on the choice of the Nash equilibrium. In the simulation, we always choose the one that is calculated first and gets reasonable results. But this can possibly change with different choices of equilibrium. We have discussed that the Nash Q-learning is guaranteed to converge only for certain types of sub-games. There is no guarantee that such games are contained in every state of our simulation. Thus at least, our agents always choose the first calculated Nash equilibrium, which consequently solves the equilibrium choice problem. However, this choice could potentially influence the behavior of agents. The results presented in Section 4 could potentially look different if the agents always choose e.g. the Nash equilibrium that yields the highest profit to them. We intentionally decided not to choose such an option because such equilibrium could be different for both agents—that is, they could consistently choose actions that the opponent would not

predict. Put simply, for Nash Q-learning it is important that the agents consistently choose the same Nash equilibrium and this requirement would likely not be fulfilled if they always chose the Nash equilibrium with highest payoff to them.

Another technical issue arises in Q-learning in general. The estimates that the agents learn are biased by initial values of the estimate [23]. In our case, agents' initial estimates are such that all games are set to have zero payoffs for each action. With too low or too quickly decreasing exploration rates, our agents do not take enough time steps to properly explore the game and usually both end up in the left edge of the street with prices set to zero. This can be explained by the initial estimates set to zero. The agents simply chose the first available Nash equilibrium from the initial estimate which is to move left and set price to zero. This way, they very quickly end up in the left corner of the street and stay there. For that reason, we set the exploration rate relatively high, which helps with better game estimates. However, our agents keep exploring even after 20,000 steps, which might cause the side-switching behavior that we discussed in Section 4.

6. Conclusion

This paper aims at examining whether simulated adaptive behavior in Hotelling's location model without perfect information brings different results from the theoretical model. Originally, Hotelling's model with linear consumer costs supported the Principle of Minimum Differentiation, where the sellers meet in the middle of the street. However, the model was shown to be invalid [4], and valid alternatives with adjusted consumer cost function have been proposed [5]. Nevertheless, the adjustments have changed the interpretation of the model. The adjusted cost function causes the sellers to differentiate. The consumers' quadratic cost function even supports the Principle of Maximum Differentiation, where the sellers locate themselves as far as possible from each other, which is basically the opposite of the original model's results.

We simulate the behavior of adapting agents innovatively guided by the Nash Q-learning algorithm in an environment under the later Hotelling's setting. Although we use a quadratic consumer cost function in the simulation, some of the results show behavior similar to the original model with linear consumer costs. If the discount rate is sufficiently high, the agents situate themselves near each other and switch positions often. As a result, their profit is lower because their local monopoly shrinks, and consequently, they have to compete in price. This behavior resembles an aggressive market strategy, where the firms differentiate as little as possible and only compete in price to eliminate the opponent from the market. Our results suggest that the Principle of Minimum Differentiation can still be justified based on repeated interactions of the agents and profit optimization in the long run. In cases with lower discount rates, agents mostly stay on opposite edges of the street, which complies with the Maximum Differentiation Principle and thus with the location model with quadratic consumer costs [4].

Aside from the simulation, a survey of relevant learning algorithms has been conducted. The Nash Q-learning among the reinforcement learning, in general, was evaluated in-depth as a suitable adaptive method for agent-based simulations with applications in Economics. We find the method well motivated for small-scale discrete economic agent-based simulations because the method does not require knowledge of the environment or the opponents. Instead, the agents repeatedly interact. Based on the interactions, they model each other and their environment and try to maximize their profits while keeping their opponent's interests in mind, which goes hand in hand with agents' behavior in classical game-theoretic models.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Jan Vainer: Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Visualization, Investigation. **Jiri Kukacka:** Conceptualization, Resources, Supervision, Writing - review & editing, Project administration, Funding acquisition.

Acknowledgements

Jiri Kukacka gratefully acknowledges financial support from the Charles University PRIMUS program [project PRIMUS/19/HUM/17] and from the Charles University UNCE program [project UNCE/HUM/035].

Appendix A. Alternative setups

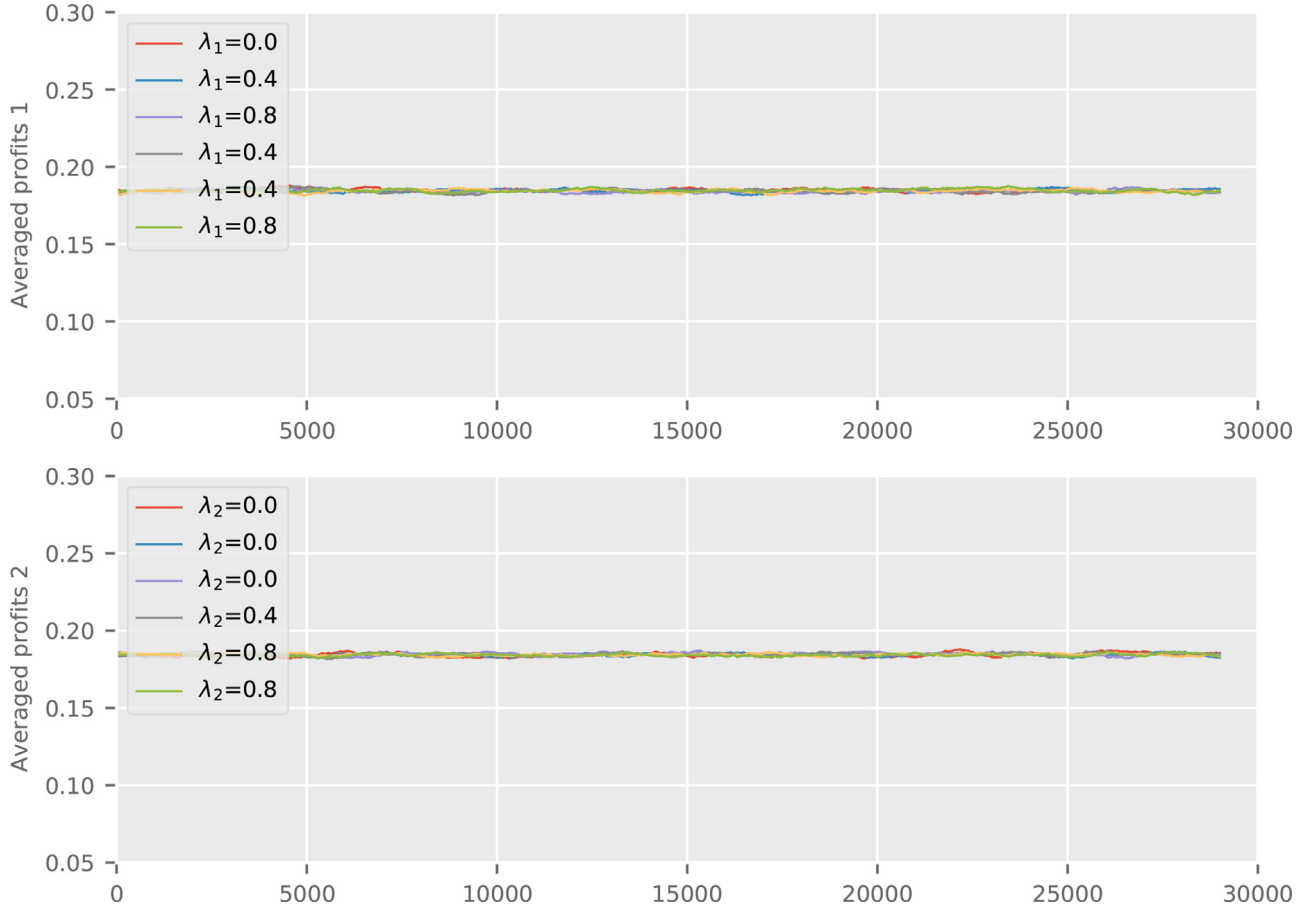
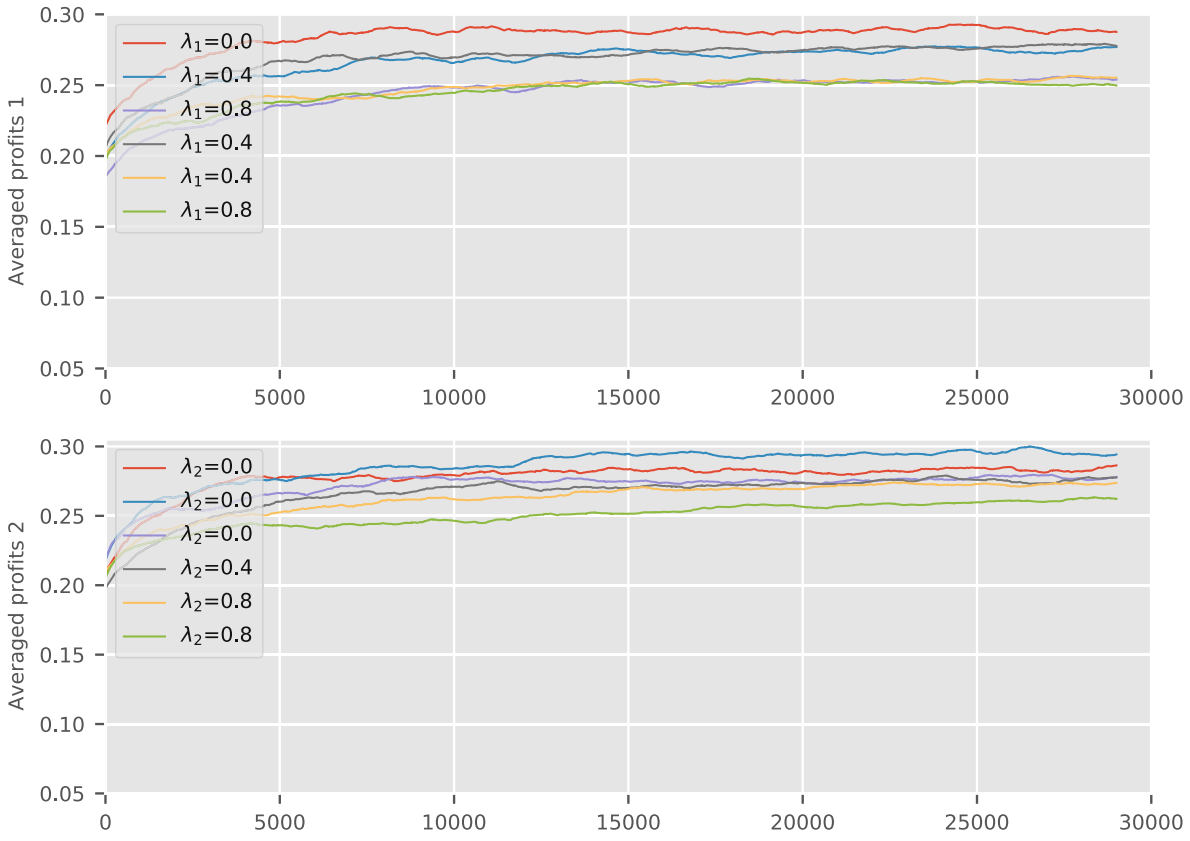


Fig. A1. Average profits for random agents. *Note:* Moving average profits for each discount rate combination is averaged across all 100 simulations at each time step. The upper sub-plot shows average profits of first agents and bottom sub-plot shows average profits of second agents. Length of rolling average is $n = 1,000$.



Note: Moving average profits for each discount rate combination is averaged across all 100 simulations at each time step. The upper sub-plot shows average profits of first agents and bottom sub-plot shows average profits of second agents. Length of rolling average is $n = 1,000$.

Fig. A2. Average profits for Q-learning agents. *Note:* Moving average profits for each discount rate combination is averaged across all 100 simulations at each time step. The upper sub-plot shows average profits of first agents and bottom sub-plot shows average profits of second agents. Length of rolling average is $n = 1,000$.

References

[1] Bester H, de Palma A, Leininger W, Thomas J, von Thadden E-L. A noncooperative analysis of Hotelling's location game. Games Econ Behav 1996;12(2):165–86. doi:10.1006/game.1996.0012.
[2] Cerboni Baiardi L, Naimzada AK. Experimental oligopolies modeling: a dynamic approach based on heterogeneous behaviors. Commun Nonlinear Sci Numer Simul 2018;58:47–61. doi:10.1016/j.cnsns.2017.05.010. Special Issue on 'Dynamic Models in Economics and Finance'
[3] Chmura T, Goerg SJ, Selten R. Learning in experimental 2x2 games. Games Econ Behav 2012;76(1):44–73. doi:10.1016/j.geb.2012.06.007.
[4] d'Aspremont C, Gabszewicz JJ, Thisse J-F. On Hotelling's "Stability in competition". Econometrica 1979;47(5):1145–50. doi:10.2307/1911955.
[5] Economides N. Minimal and maximal product differentiation in Hotelling's duopoly. Econ Lett 1986;21(1):67–71. doi:10.1016/0165-1765(86)90124-2.
[6] Friedman D. Evolutionary economics goes mainstream: a review of the theory of learning in games. Journal of Evolutionary Economics 1998;8(4):423–32. doi:10.1007/s001910050071.
[7] Gibbard A. Manipulation of voting schemes: a general result. Econometrica 1973;41(4):587–601. doi:10.2307/1914083.
[8] Golman R, Page SE. Basins of attraction and equilibrium selection under different learning rules. Journal of Evolutionary Economics 2009;20(1):49. doi:10.1007/s00191-009-0136-x.
[9] Hanaki N, Tanimura E, Vriend NJ. The principle of minimum differentiation revisited: return of the median voter. Journal of Economic Behavior & Organization 2019;157:145–70. doi:10.1016/j.jebo.2017.12.014.
[10] Hotelling H. Stability in competition. The Economic Journal 1929;39(153):41–57. doi:10.1007/978-1-4613-8905-7_4.
[11] Hu J, Wellman MP. Nash Q-learning for general-sum stochastic games. Journal of Machine Learning Research 2003;4:1039–69.
[12] Khan A, Peeters R. Imitation and price competition in a differentiated market. Journal of Economic Dynamics and Control 2017;82:177–94. doi:10.1016/j.jedc.2017.06.005.
[13] Lahkar R, Seymour RM. Reinforcement learning in population games. Games Econ Behav 2013;80:10–38. doi:10.1016/j.geb.2013.02.006.
[14] Littman ML. Friend-or-Foe Q-learning in general-sum games. ICML 2001;1:322–8.
[15] Lloyd WF. Two lectures on the checks to population. JH Parker; 1833.
[16] Matsumura T, Matsushima N, Yamamori T. Evolution of competitive equilibrium with endogenous product differentiation. Tech. Rep.. Osaka University ISER Discussion Paper No 776; 2010. doi:102139/ssrn1615862.
[17] Nagel R. Unraveling in guessing games: an experimental study. Am Econ Rev 1995;85(5):1313–26.

[18] Nagel R, Vriend NJ. An experimental study of adaptive behavior in an oligopolistic market game. Journal of Evolutionary Economics 1999;9(1):27–65. doi:10.1007/s001910050074.
[19] Nakov A, Nuño G. Learning from experience in the stock market. Journal of Economic Dynamics and Control 2015;52:224–39. doi:10.1016/j.jedc.2014.11.017.
[20] Nash JF, et al. Equilibrium points in n-person games. Proceedings of the National Academy of Sciences 1950;36(1):48–9.
[21] Pecora N, Sodini M. A heterogenous Cournot duopoly with delay dynamics: Hopf bifurcations and stability switching curves. Commun Nonlinear Sci Numer Simul 2018;58:36–46. doi:10.1016/j.cnsns.2017.06.015.
[22] Roth AE, Prasnikar V, Okuno-Fujiwara M, Zamir S. Bargaining and market behavior in Jerusalem, Ljubljana, Pittsburgh, and Tokyo: an experimental study. Am Econ Rev 1991;81(5):1068–95.
[23] Sutton RS, Barto AG. Reinforcement learning: an introduction. Cambridge, MA: MIT Press; 1998.
[24] Waltman L, Kaymak U. Q-learning agents in a Cournot oligopoly model. Journal of Economic Dynamics and Control 2008;32(10):3275–93. doi:10.1016/j.jedc.2008.01.003.
[25] Watkins CJCH. Learning from delayed rewards 1989.