UNIVERSITÄT BIELEFELD

Faculty of Business Administration and Economics

Working Papers in Economics and Management

No. 02-2023
March 2023

Implications of algorithmic wage setting on online labor platforms: a simulation-based analysis

Herbert Dawid Philipp Harting Michael Neugart

Bielefeld University
P.O. Box 10 01 31
33501 Bielefeld − Germany

ISSN 2196−2723

www.wiwi.uni−bielefeld.de

Implications of algorithmic wage setting on online labor platforms: a simulation-based analysis*

Herbert Dawid[†] Philipp Harting[‡] Michael Neugart[§]

March 2023

Abstract

We study how the use of machine-learning based algorithms for the determination of wage offers affects workers' wages on online labor platforms. Firms use reinforcement-learning to update posted wages on the platform, and heterogeneous workers send applications based on the posted information. We show that if firms use a deep Q-network (DQN), as an example of a state-of-the-art machine learning algorithm, the emerging wages closely resemble the equilibrium outcome. However, slightly changing the setup of the algorithms can lead to substantial collusion and wages well below the equilibrium level. In particular, we identify a specific property of the algorithms, namely whether *experience replay* is used, which determines whether collusion occurs or not. Our findings are robust with respect to many features of the model, including the design of the online labor platform.

Keywords: online digital labor platforms, duopsony, deep Q-network, experience replay, wages

1 Introduction

Evidence suggests that workers on online labor platforms earn low wages, sometimes even below countries' legislated minimum wages (Berg et al., 2018). One explanation has been the monopsony power of employers which would allow firms operating through online labor platforms to lower wages below workers' marginal product. Monopsony power, it is argued, may arise because there is a low number of employers posting tasks on online labor platforms, from idiosyncratic preferences that workers have over tasks, or from search frictions on the side of the employers or workers – although it is likely that online labor platforms have decreased these frictions substantially. Estimates of the labor supply elasticity to firms using data from online labor platforms arrive at elasticities as low as 0.1 (Azar et al., 2019; Dube et al., 2020; Duch-Brown et al., 2022), suggesting considerable monopsony power of employers posting tasks on these platforms.

Another explanation of low wages of workers on online labor platforms could be that employers posting tasks on these platforms use algorithms to set wages that implicitly collude to raise employers' profits at the expense of workers' wages. Could it be that the observed low wages for online platform workers are the result of employers using algorithms that collude when posting tasks?

*Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)– Project-ID 317210226 – SFB 1283. Computational resources were provided by the Center for Parallel Computing at the University of Paderborn. We are grateful for helpful comments by participants of the 2022 Conference of the Society for the Advancement of Behavioral Economics (South Lake Tahoe), the Symposium "On-The-Fly Computing" (Paderborn), the 2022 Meeting of the German Network for New Economic Dynamics, the Workshop on Computational and Experimental Economics (Vancouver) and by seminar audiences at the University of Grenoble and Bielefeld University.

[†]Department of Business Administration and Economics, Bielefeld University, hdawid@uni-bielefeld.de

[‡]Department of Business Administration and Economics, Bielefeld University, philipp.harting@uni-bielefeld.de

[§]Department of Law and Economics, Technical University of Darmstadt, neugart@vwl.tu-darmstadt.de

It is hard to get reliable data on whether employers using labor online platforms deliberately choose algorithms that result in collusive behavior. What we do know, however, is that among the firms using online platforms in general “A majority of retailers track the online prices of competitors. Two thirds of them use software programs that autonomously adjust their own prices based on the observed prices of competitors”. (European Commission, 2017, p. 5). Moreover, the same report states that “The availability of real-time pricing information may also trigger automatised price coordination. The wide-scale use of such software may in some situations, depending on the market conditions, raise competition concerns.”(p. 5). This suggests that employers may use algorithms for posting tasks on online labor platforms and it is important to understand the implications of such use of algorithms.

Which pricing algorithms firms exactly use is a black box and few attempts have been made to study collusion by algorithms empirically (Assad et al., 2022). Therefore, the possibility of algorithmic collusion has so far been explored using market simulations. In the context of product markets, several recent contributions have shown in different settings that collusion emerges if firms use a (tabular) Q-learning algorithm (Calvano, Calzolari, Denicoló and Pastorello, 2020; Calvano et al., 2021; Klein, 2021). The amount of collusion is, however, mitigated or even overall eliminated if the algorithm can incorporate additional information about a market's demand structure (Asker et al., 2022).[1]

Our analysis extends this stream of literature in two important dimensions. First, we consider the implications of the use of algorithms in the context of labor markets rather than product markets. Second, whereas the work discussed above has assumed that firms rely on basic tabular Q-learning algorithms developed in the 1990s, we employ in our analysis a state-of-the art reinforcement learning algorithm designed for the use in problems with large state spaces, in particular also discretizations of continuous state spaces. Given that firms choosing wages (or prices) face problems with continuous state spaces, it seems natural that they rely on advanced algorithms developed for that purpose. More precisely, we assume that firms use deep Q-network (DQN) algorithms, which are highly popular in the machine learning literature (Mnih et al., 2015; Goodfellow et al., 2016), for updating their wage setting rules over time. With respect to the information available for training the algorithm, we assume, similar to (Calvano, Calzolari, Denicoló and Pastorello, 2020; Calvano et al., 2021; Klein, 2021) that only wage information publicly observable on the platforms and the firms own profit is used by the algorithm.

Our economic environment consists of a duopsony model of the labor market, where two firms compete for workers by posting on an online labor platform tasks to be carried out. We distinguish between two different labor platforms designs where a key distinguishing factor is whether firms combine tasks placed on the platform with binding wage offers or not. On Amazon Mechanical Turk, which is one of the five online digital labor platforms dominating the market (see https://ilabour.oii.ox.ac.uk/onlinelabour- index/), tasks are placed as take-it-or-leave-it offers. In a take-it-or-leave-it set-up, an employer posts a task with the wage she is willing to pay. Workers then may accept this task or not at the offered wage. There is no bargaining over the wage of the job. Other prominent online digital labor platforms such as Freelancer, Guru, or Upwork follow a bidding model. Here, employers post a task, sometimes with a wage range they would be willing to pay, and workers bid for the jobs informing the employers about the wage they want to be paid. Employers then decide whom to offer the job.

The two matching protocols of online labor platforms make different sets of information public. In the take-it-or-leave-it framework, employers (and their algorithms) have access to information about

[1] Earlier work studying the interaction of algorithmic decision makers on product markets considered mostly quantity competition and the learning behavior of genetic algorithms or classifier systems. This stream of literature found no indication for the emergence of collusion but, depending on the considered setup, convergence to Nash equilibrium (Vriend, 2000) or even Walrasian equilibrium (Arifovic, 1994).

the wages their competitors pay for their tasks. This is not the case for the bidding framework, where the agreed wage is private information between the firm and the worker, whom the job was offered and who took it. Standard game-theoretic reasoning implies that collusion can be easier established and sustained as equilibrium outcome in the take-it-or-leave-it framework, since in this setting each firm can observe deviations from the collusive wage (i.e. increasing the own wage offer to attract additional workers) by the competitor and then retaliate in the following periods (Fudenberg and Maskin, 1986).

We use this economic environment to analyze the consequences of algorithmic wage setting by firms for the emerging wages in this market. First, we study whether the design of the platform has an impact on wages and on the prevalence of collusion in the market. Second, we explore the impact of a variation of the DQN algorithm. In particular, we consider a scenario where an important feature of DQNs called experience replay (see below) is disabled. We consider this variation of the algorithm, since experience replay is a crucial difference between DQN and the basic Q-learning algorithms studied in most previous work on collusion in markets.

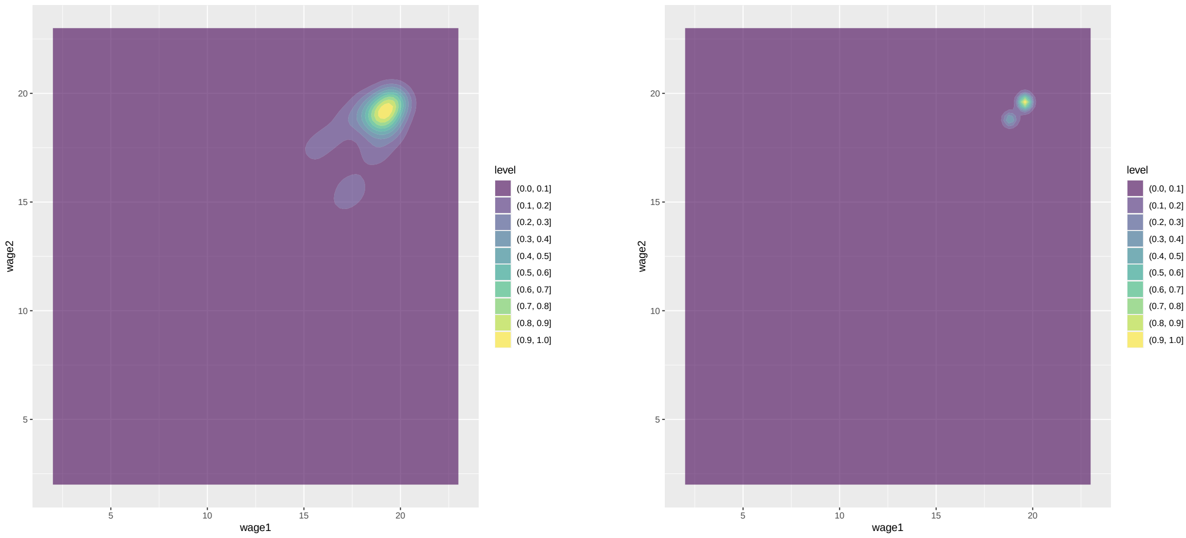
2 Interaction on the labor platform

Empirically, we observe substantial market power of employers on online digital labor platforms. For example, about 10% of employers post approximately 98% of the tasks on Amazon Mechanical Turk (Kingsley et al., 2015). To capture such market power of employers on online labor platforms, we consider a stylized duopsony model of the labor market, in which two firms compete by simultaneously posting tasks (and associated wage offers) on the platform to maximize their profits. Workers have heterogeneous preferences with respect to carrying out the tasks announced by the two firms. They decide which task to accept in light of these preferences, the wages offered by both firms as well as their reservation utility. A Nash equilibrium in the (one-shot) model is a pair of wage offers such that each firm maximizes its profit given the wage offered by the competitor. Collusive wages result if both wages are chosen in a way to maximize the sum of the profits of both firms. A crucial parameter in the model is the strength of the worker preferences between tasks, which is denoted by e. The smaller the parameter e is, the larger is the intensity of competition between firms. If the parameter e is very large, then under Nash equilibrium wages workers prefer not working to carrying out the less preferred task, and a 'local monopsony' scenario arises. In this scenario, the two firms do not directly compete with each other, but each firm effectively has its local pool of potential employees for whom it acts as monopsonist (see Appendix A for a detailed description of the model and the calculation of Nash equilibrium and collusive wages for different values of e).

3 Deep Q-Network learning of wage setting strategies

In our baseline setting, we implement the duopsony model in a simulation framework where two firms ($i = 1, 2$) every period simultaneously decide on wage offers, each using a DQN algorithm. In a DQN (Mnih et al., 2015), firm i sets its wage using a deep neural network which represents an action-value function $Q(s_i, w_i; \theta_i)$, where $s_i \in S$ is the state of firm i in period t, $w_i \in W$ a potential action of firm i in period t, and θ_i the vector of weights in the neural network.

In our setting, the set of of firms' actions consists of an equally discretized wage space $W \subset [\underline{w}, \overline{w}]$. We define states in the take-it-or-leave-it model as all possible combinations of wages chosen by the two firms in the previous period, $s_{i,t} = (w_{1,t-1}, w_{2,t-1})$. In the bidding model, since the firm does not know which wage was offered by her competitor, the states are defined only as the wage chosen by the firm under consideration in the previous period, i.e. $s_{i,t} = w_{i,t-1}$.



(a) (b)

Figure 1: Figure shows heat map for learned wages $wage1$ and $wage2$ of firms 1 and 2, respectively, over 100 repetitions at iteration 500,000. Simulation model is take-it-or-leave-it (a) and bidding (b) online labor platform with firms applying DQN with experience replay to set wages. Density estimates scaled to a maximum of one.

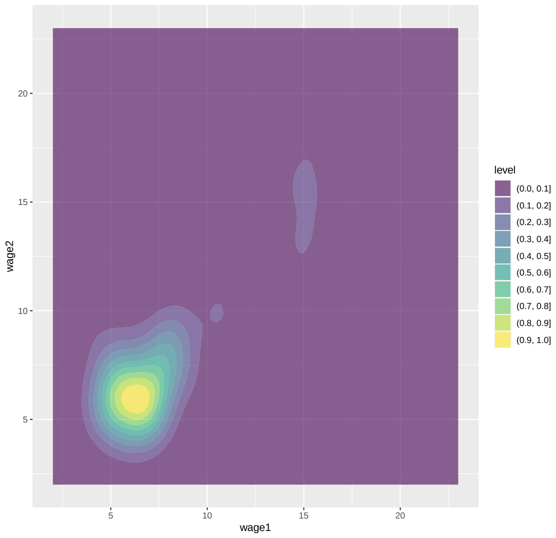
Input nodes of the DQN representing the action-value function $Q(s_i, w_i; \theta_i)$ correspond to the components of the states s_i of a firm. Each output node corresponds to an action $w \in W$ and gives the estimation of the Q-value for this action. A firm i trains the DQN by adjusting every period the vector of weights θ_i in order to reduce the mean-squared error of the right hand side of an appropriate Bellman equation (see the pseudo code in Algorithm 1). The adjustment of the weights follows a standard gradient descent algorithm. An important feature of a DQN is that in each updating period it is trained using not only the most recent observation but a sample from the set of past observations. This feature is called experience replay and has been shown to foster convergence and performance of the algorithm (Mnih et al., 2015). In contrast, standard tabular Q-learning algorithms, which have been used in almost all previous market simulation studies, rely only on the most recent observation in each step of their training.

A greedy algorithm guarantees that there is experimentation about the actions chosen. The ϵ-greedy model of exploration chooses the action $w_{i,t}$ maximizing $Q(s_{i,t}, w_i; \theta_{i,t})$ with probability $1 - \epsilon_t$ and randomizes uniformly over all actions with probability ϵ_t. The exploration rate declines with $\epsilon_t = e^{-\beta t}$, where $\beta > 0$ is a parameter. Thus, initially, firms choose their wage offers randomly, but, as time passes, actions in line with the highest Q-value become more likely.

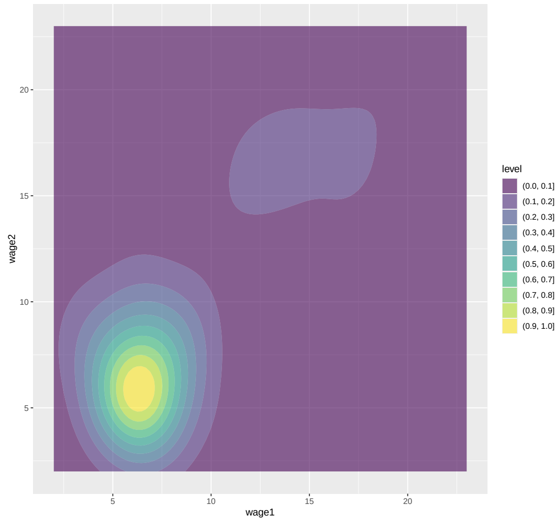
A more detailed description, including a pseudo-code, of the DQN and how it is embedded in the economic environment, is given in Appendix B.

4 Results

We simulate the model with a baseline parameter setting (see Appendices A and B), but vary parameters for robustness checks. The parameters chosen imply a Nash equilibrium in wages of 19.6 and a collusive wage of 5.2 for the analytical model (see Appendix A). This is the benchmark to which we compare the simulation in which firms set wages using a DQN with or without experience replay. A single run consists of 500,000 iterations. For each parameter setting, we conduct 100



(a)



(b)

Figure 2: Figure shows heat map for learned wages $wage1$ and $wage2$ of firms 1 and 2, respectively, over 100 repetitions at iteration 500,000. Simulation model is take-it-or-leave-it (a) and bidding (b) online labor platform with firms applying DQN without experience replay to set wages. Density estimates scaled to a maximum of one.

runs.

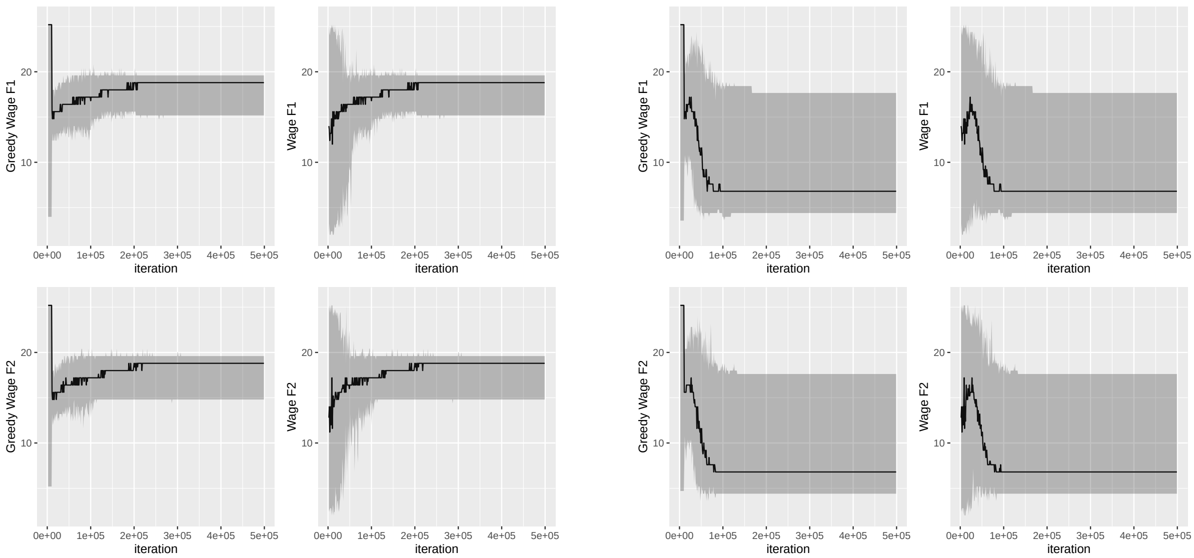
To evaluate the learning ability of the algorithm without interacting firms, we first consider the case where the strength of worker preference parameter e is so large that we are in the local monopsony scenario. As is shown in Figure 5 in Appendix C, in this scenario both firms learn to choose the wage which maximizes their profit regardless of the platform design and for both DQN variants with and without experience replay.

Next we consider the duopsony model for scenarios where the worker preference parameter is sufficiently small such that there is direct competition between the two firms. We start with firms using a DQN with experience replay to learn wages, and apply the algorithm to the take-it-or-leave-it platform and the bidding platform. Afterwards, we explore the DQN without experience replay for both online labor platforms.

Figure 1 shows the heat map of wages for 100 repetitions for the take-it-or-leave-it (panel (a)) and for the bidding (panel (b)) online labor platform. With both firms using a DQN with experience replay they do not learn to collude on either online labor platform. For both platform designs wages converge to values close to the Nash equilibrium in a vast majority of runs, where the variance of emerging wages across runs is slightly larger under the take-it-or-leave-it platform design.

While firms applying a DQN with experience replay to set wages do not collude, collusion over wages emerges if firms use a DQN with experience replay switched off, see Figure 2. For both online labor platform designs firms in all runs end up setting wages substantially below the Nash equilibrium wages, and in most runs the emerging wages are very close to the collusive wage of $w_1^C = w_2^C = 5.2$. As in the case with experience replay, the effect of the platform design on wage levels is minor. But regardless of the considered platform, the difference in the emerging wage between the cases where firms use DQN with and without experience replay is striking and highly significant.

The two panels in Figure 3 show the dynamics of the distribution across runs of greedy wages (i.e. the wages maximizing the firm's Q-function) and actual wages (which might be the greedy



(a) (b)

Figure 3: Figure shows greedy wages and wages of firms 1 and 2 over time of simulation for 100 repetitions. Simulation model is take-it-or-leave-it online labor platform with firms applying DQN with experience replay (a) and without experience replay (b) to set wages. Black line is median. Upper and lower limit of gray area are 25% and 75% quantiles, respectively.

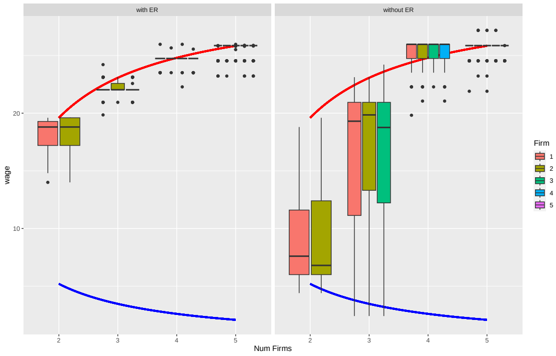
wage or a random wage) over time for the take-it-or-leave-it online labor platform. With experience replay, Nash wages are learned after approximately 200,000 iterations. Without experience replay, collusive wages are learned after approximately 100,000 iterations. Consistently with Figures 1(a) and 2(a), these figures highlight that the variance of emerging wages across runs is substantially larger without than with experience replay. In particular, without experience replay, apart from the large majority of runs, which end up close to the fully collusive wage level, there is a small but positive fraction of runs in which wages emerge, which exhibit partial collusion in the sense that they are clearly below the Nash equilibrium wages but also substantially above the collusive wage. For the bidding model, we get similar results, see SI Appendix.

5 Robustness

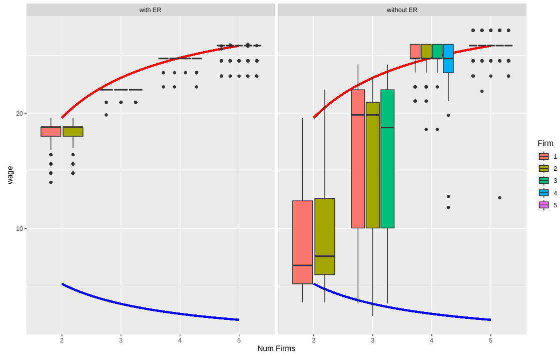
How robust are our findings, or asking differently, are there other ways for firms to exploit algorithmic collusion, apart from using algorithms without experience replay? To explore this question, we ran various robustness tests.

Figure 4 shows how wage setting changes as more firms are in the market for both online labor platforms, respectively. Wages of all firms on the market set by their DQN with (left panel) and without (right panel) experience replay are plotted with the number of firms increasing from two to five. The Whisker-box-plots show the learned wages for 100 repetitions after 500,000 iterations. They are compared to the analytical solutions for a duopsony or oligopsonies. Blue lines indicate collusive wages and red lines Nash wages. While for the DQN with experience replay outcomes are close to Nash wages regardless of the number of firms competing on the platform, collusion for the DQN without experience replay breaks down as the number of competitors grows. If four or five firms are active on the platform, wages converge to a level close to Nash equilibrium also if firms use a DQN without experience replay.

While collusion is sensitive to the number of firms in the market, i.e. competitive pressure,



(a)



(b)

Figure 4: Figure shows wages of all firms with the number of firms in the market varying from two to five on a take-it-or-leave-it (a) and bidding (b) online labor platform with firms using a DQN with experience replay (left panel) and without experience replay (right panel). Red lines indicate Nash wages and blue lines indicate collusive wages.

several other parameter variations yield robust results. We made tasks offered by the firms more or less heterogeneous from the perspective of the workers, i.e. varied (e), which may also be interpreted as a measure of competition in the market given the number of firms. We made the environment in which firms interact more uncertain by introducing a stochastic labor productivity. We also checked for asymmetric firms. Furthermore, we explored the effect on outcomes of various parameters relating to the DQN including the learning rate used in the gradient descent, β, i.e. the speed with which exploration decays, the number of hidden layers, memory size, and the frequency of updating of network weights. Our finding that firms do not collude if the algorithm used for wage setting learns on a broader set of past observations, i.e. uses experience replay, stays robust. It is also a robust result that the algorithm with no experience replay supports collusive behavior. Detailed results of these robustness tests are in Appendix C.

6 Discussion

Collusion of firms on online labor platforms is a possible outcome that may be achieved as firms use algorithms that do not train on a broader set of past outcomes. Should one be concerned that algorithms are used, which lead to wages below the equilibrium level on online labor platforms?

Large traffic on online labor platforms suggests that machine learning algorithms have the potential for substantial cost reductions in hiring. They are also faster in making decisions. Thus, on the one hand, they reduce transaction costs and make matching in labor markets more efficient. On the other hand, we show that there is the possibility of implicit algorithmic collusion lowering workers' wages. However, forbidding algorithmic wage setting to avoid the detrimental effects on workers would throw the baby out with the bathwater. More so, as we show that algorithms using experience replay are actually not colluding and more competition moves wages closer to Nash wages when a DQN without experience replay sets wages. When government agencies are facing obstacles to increase competition on online labor platforms, a conclusion from our analysis is that it has to be checked whether the algorithms are apt to collusion. Our findings provide a clear guidance on which properties of the algorithm are crucial in this respect. Firms using algorithms that during the training phase update their parameters (i.e. the weights in the Q-network) based on a broader set of past observations are less prone to implicit collusion. The dependence of algorithmic collusion on features of these algorithms and how they interact with the market environment make it, however, a very challenging task for anti-trust agencies to regulate firms, see also (Calvano, Calzolari, Denicoló, Harrington Jr and Pastorello, 2020).

Compared to actual online labor platforms our model might be rather simplistic. Nevertheless, we are able to show that there is the possibility for wage cutting as firms delegate posting tasks and wages on online labor platforms to algorithms. While there is certainly a lot more to do with respect to analyzing how particular algorithms operate in different institutional settings, our analysis highlights that new forms of employment create new challenges for policy concerned with functioning labor market competition and fair distribution of income.

Appendix

A Duopsony Model

In our analytical formulation of the platform economy, we have two firms ($i = 1, 2$) being located on the opposite sides of a Salop circle (Salop, 1979) with circumference 1. The two firms set wages w_1 and w_2 simultaneously to maximize their profits. Profit of firm i is $\pi_i = A_i L_i - w_i L_i$, where L_i is the quantity of labor employed by firm i, and $A_i > 0$ is a productivity parameter. In total, there is a fixed number N of workers uniformly distributed on the Salop circle. A worker with location $x^+ \in [0, 1/2]$ on the Salop circle has utility $u_i(x^+; w_{i,t}) = w_{i,t} - e \cdot d_i(x^+)$ from accepting the task offered by firm i at a wage of $w_{i,t}$. The function $d_i(x^+)$ measures the disutility of the worker from carrying out the task offered by firm i, where $d_1(x^+) = Nx^+$ and $d_2(x^+) = N(1/2 - x^+)$. Utility of not working is normalized to zero, such that the worker accepts the task offered by firm i if $u_i(x^+; w_{i,t}) \geq \max[0, u_j(x^+; w_{j,t})]$. We denote the location of the worker, who for given wage offers is indifferent between accepting task of firm 1 and firm 2, as $x^{ind}(w_1, w_2)$, i.e. $u_1(x^{ind}; w_1) = u_2(x^{ind}; w_2)$. To determine the Nash equilibrium wages, we focus on the case where e is sufficiently small such that in equilibrium the indifferent worker has a strictly positive utility. In this case the labor supply to firm i is given by $L_i = (w_i - w_j)/e + N/2$ (for simplicity we abstract in our analytical calculations from the fact that in the simulation the quantity of labor is always an integer). Inserting the labor supplies into the profit functions and maximizing with respect to the wage of the firm, yields the firms' reactions functions. The intersection of the reaction functions yields the Nash equilibrium wages $w_i^* = (2A_i + A_j)/3 - Ne/2$. Checking the condition that $u_i(x^{ind}; w_i^*) \geq 0$ shows that this wage profile is a Nash equilibrium for all $e \leq \tilde{e} = 2(A_1 + A_2)/(3N)$.

Collusive wages are determined by choosing w_1, w_2 in order to maximize $\pi_1 + \pi_2$. Again, focusing on the case of small values of e, it is easy to see that under collusive wages w_1^C, w_2^C we must have $u_i(x^{ind}; w_i^C) = 0$ and from this it is directly obtained that $w_i^C = (A_i - A_j + eN)/4$. These wages maximize the sum of profits for $e \leq \tilde{\tilde{e}} = (A_1 + A_2)/N$. For values of e above this threshold, it is optimal to set wages such that some workers prefer not to work.

In our baseline specification, we choose parameters $A_1 = A_2 = 30$, $N = 104$, $e = 0.2$, $\beta = 6 \cdot 10^{-5}$, $\underline{w} = 2$ and $\overline{w} = 23.6$ in steps of 0.8. For this parameter setting $e < \tilde{e} = 0.38 < \tilde{\tilde{e}} = 0.58$ and we obtain Nash equilibrium wages of $w_1^* = w_2^* = 19.6$ and collusive wages of $w_1^C = w_2^C = 5.2$.

B Details of the DQN in the duopsony environment

Q is the main neural network for firm i with weight vector θ_i. This network is used to determine the greedy wages, i.e. the action w_i which in a state s_i is associated with the largest Q-value, and updated every period based on experience replay. $\tilde{Q}$ is the target neural network with weight vector $\tilde{\theta}$. Every τ^T periods, Q is cloned and the resulting network $\tilde{Q}$ with fixed $\tilde{\theta}$ is used for generating the Q-learning targets y_j for updating network Q. $\mathcal{D}_{i,t}$ is the array of experiences, used for experience replay in period t:

$$\mathcal{D}_{i,t} = \{(s_{i,t-\mathcal{T}}, w_{i,t-\mathcal{T}}, \pi_{i,t-\mathcal{T}}, s_{i,t-\mathcal{T}+1}), ..., (s_{i,t}, a_{i,t}, \pi_{i,t}, s_{i,t+1})\}$$

Parameters are: τ^T: Frequency of updates of weights in target neural network $\tilde{Q}$; $\mathcal{T}$: Memory size of backward window $\mathcal{D}_{i,t}$ for experience replay; n^m: Size of minibatch sample in experience replay.

Hyper-parameters of the neural network are set as follows, see also Bengio (2012): learning rate $\alpha = 0.01$, speed with which exploration decays $\beta = 6 \cdot 10^{-5}$, memory size of backward window is $\mathcal{T} = 10^5$, frequency of network update is $\tau^T = 10^4$, minibatch size $n^m = 32$, bias is zero, weight

Algorithm 1 Firm behavior

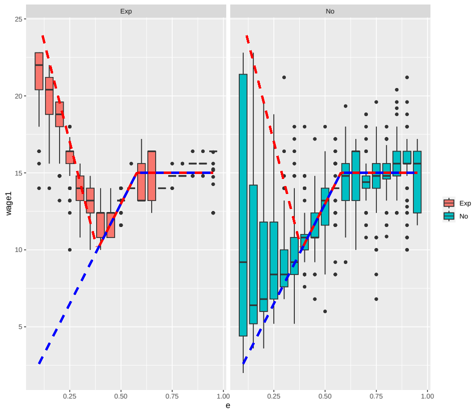
procedure DETERMINESTATE
  $s_{i,t} = (w_{i,t-1}, w_{j,t-1})$ *for take-it-or-leave-it design*
  $s_{i,t} = w_{i,t-1}$ *for bidding design*
procedure WAGEOFFER
  if $rnd < \epsilon_t$ **then** $w_{i,t} \leftarrow$ *random choice*
  else $w_{i,t} = \max_w Q(s_{i,t}, w; \theta_{i,t})$
procedure HIRING
  Workers choose whether to accept wage offer $\rightarrow L_{i,t}$
procedure PROFIT
  $\pi_{i,t} = A_i L_{i,t} - w_{i,t} L_{i,t}$
procedure UPDATEQ
  Add experience $(s_{i,t}, w_{i,t}, \pi_{i,t}, s_{i,t+1})$ *to* $\mathcal{D}_{i,t-1}$
  Drop $(s_{i,t-\mathcal{T}}, w_{i,t-\mathcal{T}}, \pi_{i,t-\mathcal{T}}, s_{i,t-\mathcal{T}+1})$ *from* $\mathcal{D}_{i,t-1}$
  $\rightarrow$ *backward window* $\mathcal{D}_{i,t}$
  Sample minibatch $\mathcal{D}_t^m$ *of size* n^m *from* $\mathcal{D}_t$
  For each $\{(s_{i,l}, w_{i,l}, \pi_{i,l}, s_{i,l+1})\} \in \mathcal{D}_{i,t}^m$
  Set $y_{i,l} = \pi_{i,l} + \gamma \max_{w'} \tilde{Q}(s_{i,l+1}, w'; \tilde{\theta}_{i,t})$
  and $L_{i,l} = (y_{i,l} - Q(s_{i,l}, w_{i,l}; \theta_{i,t}))^2$
  Average gradient: $\bar{L}'_{i,t} = 1/n^m \sum_{l \in \mathcal{D}_t^m} \frac{\partial L_{i,l}}{\partial \theta}$
  $\theta_{i,t+1} \leftarrow$ *gradient descent step using* $\bar{L}'_{i,t}$
procedure UPDATE$\tilde{Q}$
  if $t \mod \tau^T = 0$ **then** $\tilde{\theta}_{i,t+1} \leftarrow \theta_{i,t+1}$
procedure UPDATEGREEDYPARAMETER
  $\epsilon_{t+1} \leftarrow \epsilon_t e^{-\beta}$

initialization is Gaussian with mean zero and standard deviation $\sqrt{1/k}$, where k is the number of input nodes. The number of hidden layers equals two. For the bidding model the first layer has $k = 1$ and for the take-it-or-leave-it model the first layer has $k = 2$. The number of output nodes is equal to the grid size of wages. The number of nodes of the first hidden layer equals $2/3$ times the sum of input and output nodes. The number of nodes for the second layer equals $3/2$ times the sum of input and output nodes. Activation functions: logistic for hidden layers, identity for output layer. Inputs are normalized to the interval $[0, 1]$, rewards are scaled by a factor of 10^{-3}. In the runs with experience replay switched off we set $\mathcal{T} = n^m = 1$.

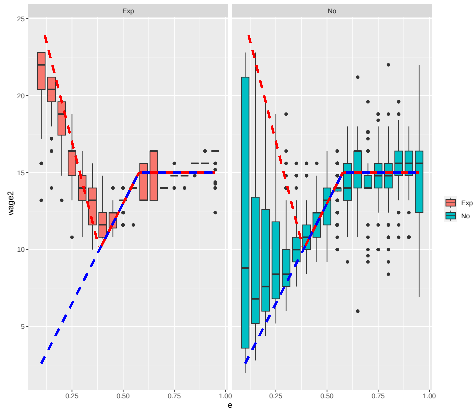
C Robustness

C.1 Strength of worker preference and monopsony wage

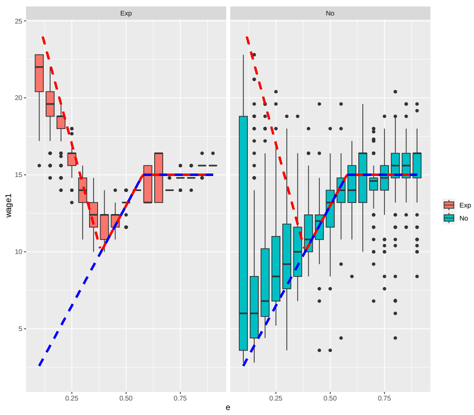
Figure 5: Effect of strength of worker preference (e) on wages



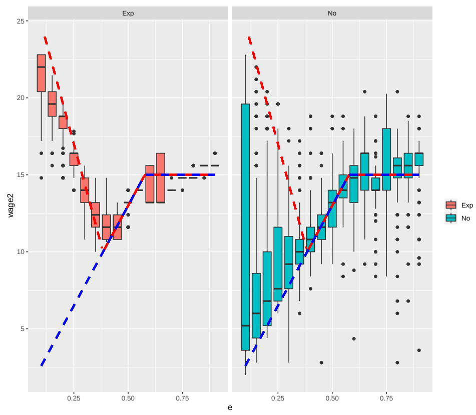
(a)



(b)



(c)

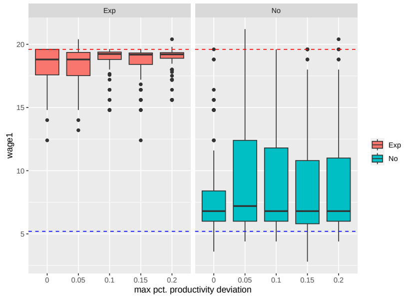


(d)

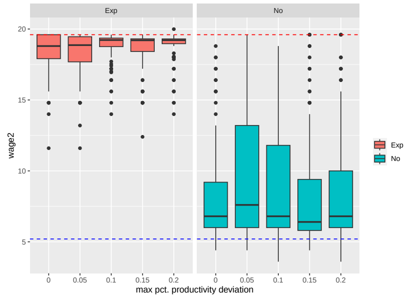
Notes: Figure shows Whisker-box-plots for wages of firms 1 and 2 as a function of the strength of worker preferences (e). Upper row shows the cases for the take-it-or-leave-it online labor platform. Lower row shows the cases for the bidding online labor platform. Red dashed lines are the Nash wages and blue dashed lines are the collusive wages according to the analytical solution. For $e \geq \widetilde{e} = 0.38$ we are in the local monopsony scenario and the Nash equilibrium wages coincide with the collusive wages (see Appendix A). For $e \geq \widetilde{\widetilde{e}} = 0.58$ it is optimal for the firms to choose wages such that some workers prefer not to work and optimal wages are given by $w_i^* = w_i^C = A_i/2$. The figure shows that DQN with experience replay persistently yields wages close to the Nash equilibrium wages, whereas without experience replay they are close to collusive wages. For $e \geq \widetilde{e} = 0.38$, i.e. the local monopsony case for which firms do not directly compete for workers, the DQN finds the monopsony wage for both online labor platforms with and without experience replay.

C.2 Stochastic environment

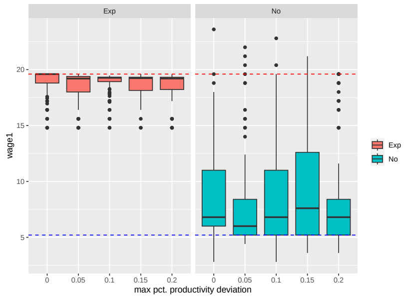
Figure 6: Stochastic labor productivity (A)



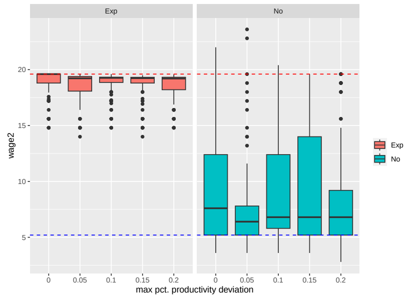
(a)



(b)



(c)

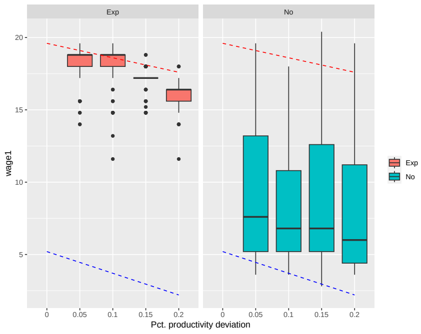


(d)

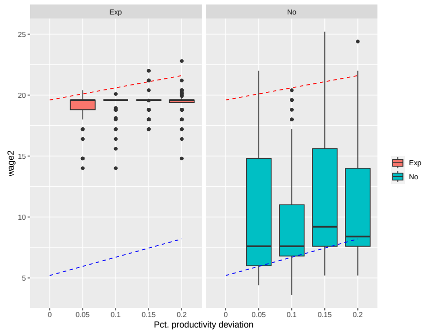
Notes: Figure shows Whisker-box-plots for wages of firm 1 (left) and 2 (right), respectively. Labor productivity $A_1 = A_2 = A$ is drawn from a uniform distribution varying the upper and lower limit of the maximum deviation from a mean (and baseline) of $A = 30$. Upper row shows the cases for the take-it-or-leave-it online labor platform. Lower row shows the cases for the bidding online labor platform. Red Whisker-boxes relate to algorithmic wage setting with experience replay. Green Whisker-boxes relate to algorithmic wage setting without experience replay. Dashed red lines and dashed blue lines indicate Nash wages and collusive wages of analytic model, respectively. We get Nash wages in the settings with experience replay and collusion in the settings without experience replay.

C.3 Asymmetric productivities

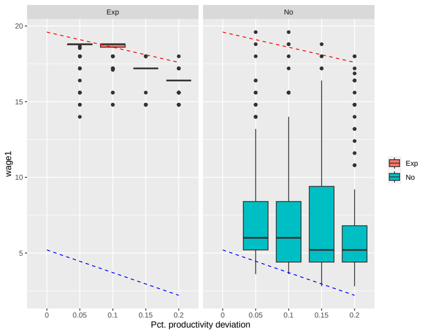
Figure 7: Asymmetric productivities



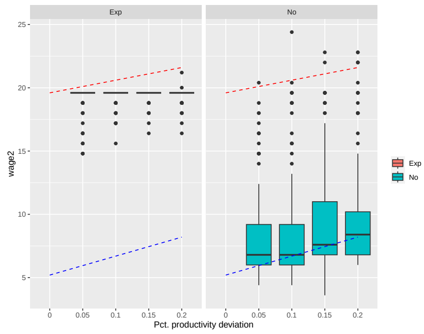
(a)



(b)



(c)

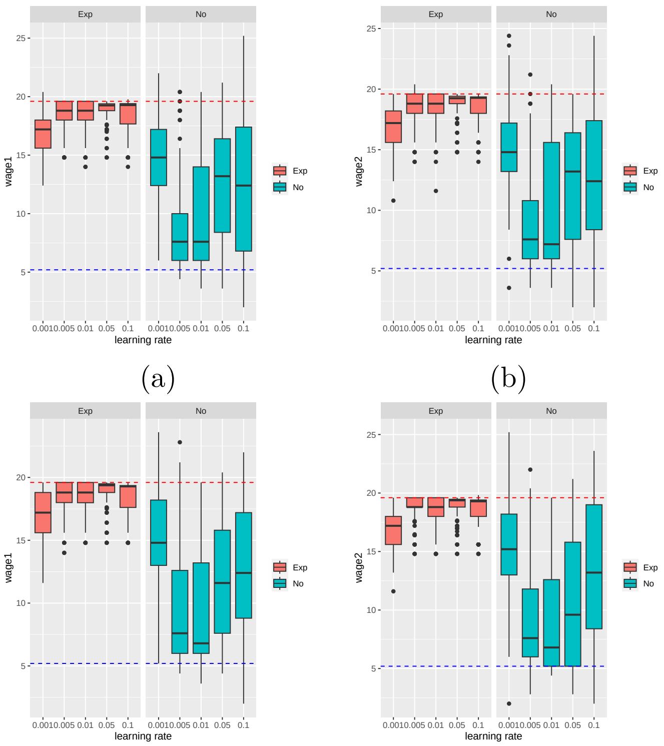


(d)

Notes: Figure shows Whisker-box-plots for wages of firm 1 (left) and 2 (right), respectively. Labor productivity of the two firms is given by $A_1 = (1-q) \cdot 30$, $A_2 = (1+q) \cdot 30$ with q varying between 0 and 0.2 (our baseline in the paper corrseponds to $q = 0$). Upper row shows the cases for the take-it-or-leave-it online labor platform. Lower row shows the cases for the bidding online labor platform. Red Whisker-boxes relate to algorithmic wage setting with experience replay. Green Whisker-boxes relate to algorithmic wage setting without experience replay. Dashed red lines and dashed blue lines indicate Nash wages and collusive wages of analytic model, respectively. We get Nash wages in the settings with experience replay and collusion in the settings without experience replay.

C.4 Learning rate

Figure 8: Learning rate



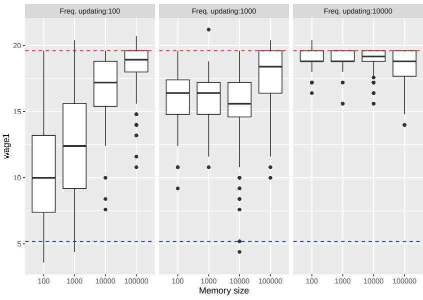
(a) (b)

(c) (d)

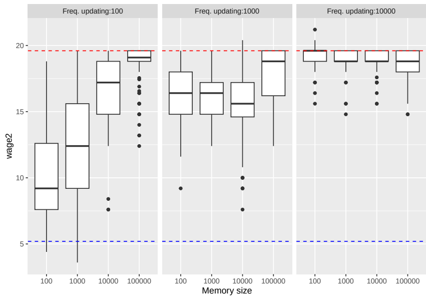
Notes: Figure shows Whisker-box-plots for wages of firm 1 (left) and 2 (right), respectively. Learning rate α of the gradient descent is varied from 0.001 to 0.1 (baseline is $\alpha = 0.01$). Upper row shows the cases for the take-it-or-leave-it online labor platform. Lower row shows the cases for the bidding online labor platform. Red Whisker-boxes relate to algorithmic wage setting with experience replay. Green Whisker-boxes relate to algorithmic wage setting without experience replay. Dashed red lines and dashed blue lines indicate Nash wages and collusive wages of analytic model, respectively. We get Nash wages in the settings with experience replay and below Nash wages without experience replay.

C.5 Memory size

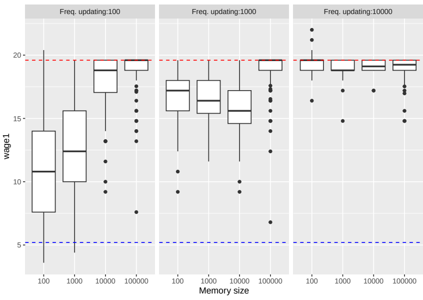
Figure 9: Memory size ($\mathcal{T}$) and frequency of updating $\left(\tau^{T}\right)$



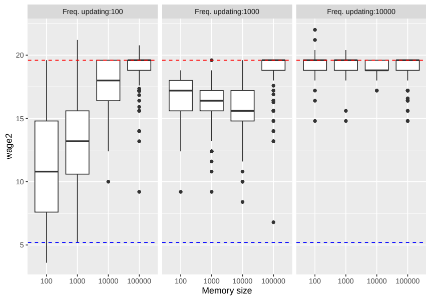
(a)



(b)



(c)

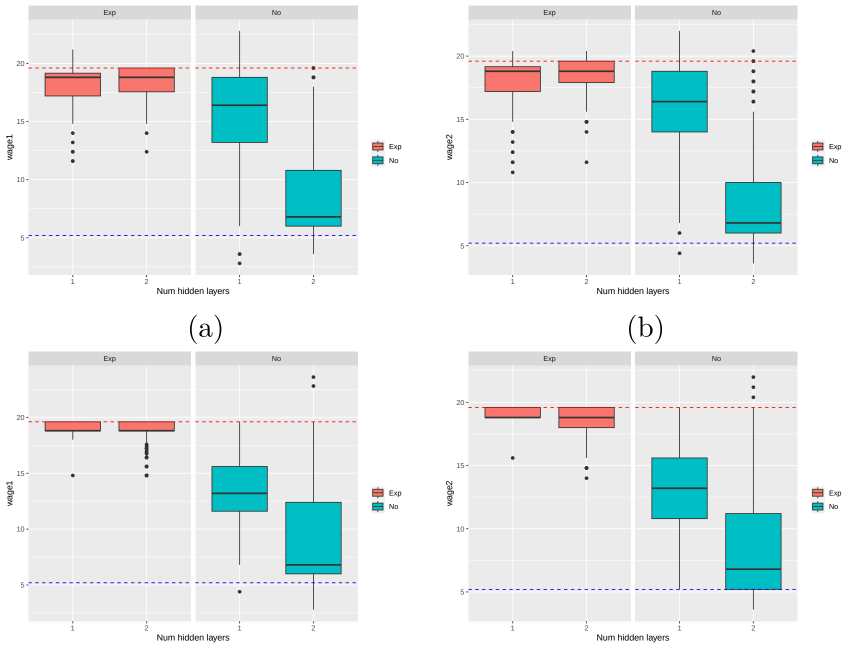


(d)

Notes: Figure shows Whisker-box-plots for wages of firm 1 (left) and 2 (right), respectively. Memory size and frequency of updating is varied (baselines are 100,000 for memory size and 10,000 for frequency update). Upper row shows the cases for the take-it-or-leave-it online labor platform. Lower row shows the cases for the bidding online labor platform. Dashed red lines and dashed blue lines indicate Nash wages and collusive wages of analytic model, respectively. We get Nash wages for high frequencies of updating and below Nash wages for low frequencies of updating if memory size is low. Wages are always well above the collusive wage.

C.6 Hidden layers

Figure 10: Hidden layers



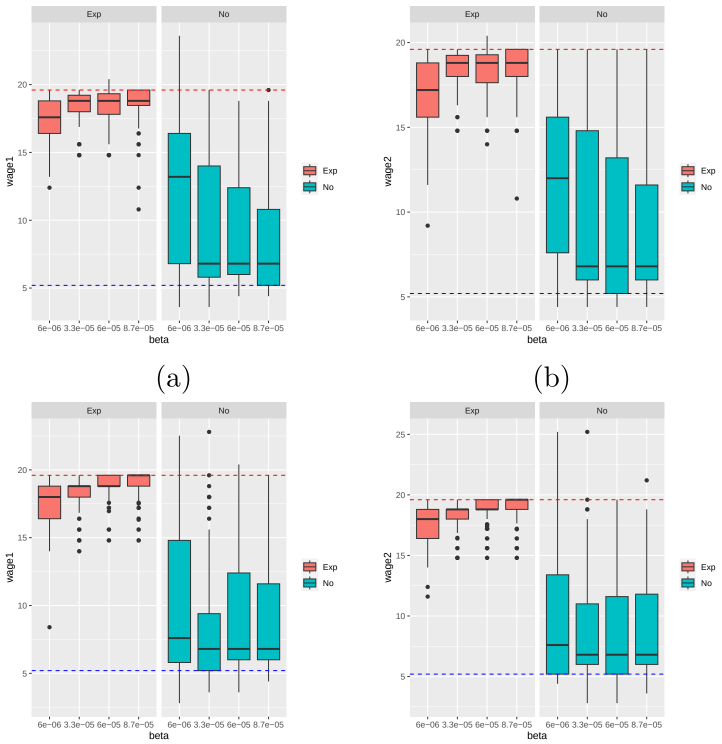
(a) (b)

(c) (d)

Notes: Figure shows Whisker-box-plots for wages of firm 1 (left) and 2 (right), respectively. Number of hidden layers is altered (baseline is two). Upper row shows the cases for the take-it-or-leave-it online labor platform. Lower row shows the cases for the bidding online labor platform. Red Whisker-boxes relate to algorithmic wage setting with experience replay. Green Whisker-boxes relate to algorithmic wage setting without experience replay. Dashed red lines and dashed blue lines indicate Nash wages and collusive wages of analytic model, respectively. We get Nash wages in the settings with experience replay and below Nash wages without experience replay. With two hidden layers algorithmic wages are closer to collusive wage .

C.7 β – speed with which exploration decays

Figure 11: β – speed with which exploration decays



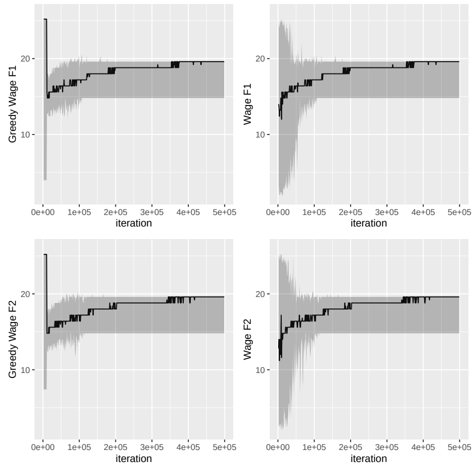
(a) (b)

(c) (d)

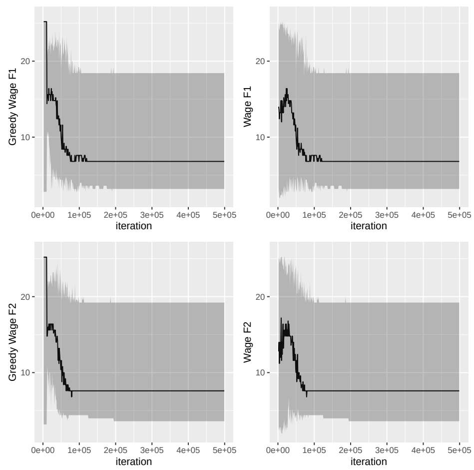
Notes: Figure shows Whisker-box-plots for wages of firm 1 (left) and 2 (right), respectively. Speed with which exploration decays is varied (baseline is $\beta = 6 \cdot 10^{-5}$). Upper row shows the cases for the take-it-or-leave-it online labor platform. Lower row shows the cases for the bidding online labor platform. Red Whisker-boxes relate to algorithmic wage setting with experience replay. Green Whisker-boxes relate to algorithmic wage setting without experience replay. Dashed red lines and dashed blue lines indicate Nash wages and collusive wages of analytic model, respectively. We get Nash wages in the settings with experience replay and collusion without experience replay.

C.8 Time series for bidding model

Figure 12: Time series for greedy wage and actual wage (bidding model)



(a)



(b)

Notes: Figure shows greedy wages and actual wages of firms 1 and 2 over duration of simulation for 100 repetitions. Simulation model is bidding online labor platform with firms applying DQN with experience replay (a) and without experience replay (b) to set wages. Black line is median. Upper and lower limit of gray area are 25% and 75% quantiles, respectively. We observe that the algorithms learn Nash wages after approximately 350,000 iterations and collusive wages after approximately 100,000 iterations.

Data Availability

Code has been deposited in GitHub. `https://github.com/ETACE/AI_Labormarket`

References

Arifovic, J.: 1994, Genetic algorithm learning and the cobweb model, *Journal of Economic Dynamics and Control* **18**, 3–28.

Asker, J., Fershtman, C. and Pakes, A.: 2022, Artificial intelligence, algorithm design, and pricing, *AEA Papers and Proceedings* **112**, 452–56.

Assad, S., Clark, R., Ershov, D. and Xu, L.: 2022, Identifying algorithmic pricing technology adoption in retail gasoline markets, *AEA Papers and Proceedings* **112**, 457–60.

Azar, J., Berry, S. and Marinescu, I. E.: 2019, Estimating labor market power, *SSRN Scholarly Paper ID 3456277*, Social Science Research Network.

Bengio, Y.: 2012, Practical recommendations for gradient-based training of deep architectures, *Neural networks: Tricks of the trade*, Springer, pp. 437–478.

Berg, J., Furrer, M., Harmon, E., Rani, U. and Silberman, M. S.: 2018, Digital labour platforms and the future of work, *Towards Decent Work in the Online World. Rapport de l'OIT* .

Calvano, E., Calzolari, G., Denicoló, V., Harrington Jr, J. E. and Pastorello, S.: 2020, Protecting consumers from collusive prices due to AI, *Science* **370**(6520), 1040–1042.

Calvano, E., Calzolari, G., Denicoló, V. and Pastorello, S.: 2020, Artificial intelligence, algorithmic pricing, and collusion, *American Economic Review* **110**(10), 3267–97.

Calvano, E., Calzolari, G., Denicoló, V. and Pastorello, S.: 2021, Algorithmic collusion with imperfect monitoring, *International Journal of Industrial Organization* **79**, 102712.

Dube, A., Jacobs, J., Naidu, S. and Suri, S.: 2020, Monopsony in online labor markets, *American Economic Review: Insights* **2**(1), 33–46.

Duch-Brown, N., Gomez-Herrera, E., Mueller-Langer, F. and Tolan, S.: 2022, Market power and artificial intelligence work on online labour markets, *Research Policy* **51**(3), 104446.

European Commission: 2017, *Report from the Commission to the Council and the European Parliament, Final report on the E-commerce Sector Inquiry*, European Commission COM(2017) 229 final, Brussels.

Fudenberg, D. and Maskin, E.: 1986, The folk theorem in repeated games with discounting or with incomplete information, *Econometrica* **54**, 533–554.

Goodfellow, I., Bengio, Y. and Courville, A.: 2016, *Deep learning*, MIT press, Camgridge, Massachusetts.

Kingsley, S. C., Gray, M. L. and Suri, S.: 2015, Accounting for market frictions and power asymmetries in online labor markets, *Policy & Internet* **7**(4), 383–400.

Klein, T.: 2021, Autonomous algorithmic collusion: Q-learning under sequential pricing, *The RAND Journal of Economics* **52**(3), 538–558.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al.: 2015, Human-level control through deep reinforcement learning, *Nature* **518**(7540), 529–533.

Salop, S. C.: 1979, Monopolistic competition with outside goods, *The Bell Journal of Economics* **10**(1), 141–156.

Vriend, N. J.: 2000, An illustration of the essential difference between individual and social learning, and its consequences for computational analyses, *Journal of Economic Dynamics and Control* **24**, 1–19.